ESP-FAQ Handbook

Release master
Espressif Systems
Apr 07, 2026

Table of contents

Table of contents i

1 Instruction 3
- 1.1 Question search . . . 3
  - 1.1.1 Question Search Techniques . . . 3
  - 1.1.2 Question Classification Categories . . . 4
- 1.2 Document contribution . . . 4
  - 1.2.1 Commit process . . . 4
  - 1.2.2 Create a new branch . . . 4
  - 1.2.3 Branch naming conventions . . . 5
  - 1.2.4 Q&A Guidelines . . . 5
  - 1.2.5 Local build environment . . . 6
  - 1.2.6 Commit message standards . . . 7
  - 1.2.7 Submit a merge request . . . 7

2 Development environment 9
- 2.1 IDE plugins . . . 9
  - 2.1.1 How to add ESP32 development board on Arduino IDE? . . . 9
  - 2.1.2 When using the Arduino IDE development platform, how to read the MAC address of the Wi-Fi that comes with ESP32? . . . 9
  - 2.1.3 How to use the flash download tool to flash the bin file generated by Arduino to ESP32? . . . 9
  - 2.1.4 How to update the esp32-arduino library version? . . . 10
  - 2.1.5 Can ESP32-SOLO-1 be developed using the Arduino software? . . . 10
- 2.2 Debugging . . . 10
  - 2.2.1 What is the serial port name of ESP devices? . . . 10
  - 2.2.2 How to block debugging messages sent through UART0 by default in ESP32? . . . 10
  - 2.2.3 How to modify the default method of RF calibration in ESP32? . . . 11
  - 2.2.4 How to modify the default method of RF calibration in ESP8266? . . . 11
  - 2.2.5 How to troubleshoot in ESP32 Boot mode? . . . 11
  - 2.2.6 When debugging with ESP32 JLINK, an ERROR occurs as: No Symbols For Freertos. How can I resolve such issue? . . . 12
  - 2.2.7 How to monitor the free space of the task stack? . . . 12
  - 2.2.8 Is it possible to use JTAG to debug with ESP32-S2? . . . 12
  - 2.2.9 How to modify the log output without changing the output level of menuconfig? . . . 12
  - 2.2.10 ESP8266 enters boot mode (2,7) and hits a watchdog reset. What could be wrong? . . . 12
  - 2.2.11 When using the ESP-WROVER-KIT board with OpenOCD, an error occurred as `Can't find board/esp32-wrover-kit-3.3v.cfg`. How can I resolve such issue? . . . 12
  - 2.2.12 The RTC_watch_dog keeps resetting during ESP32 SPI boot. What could be the reason? . . . 13
  - 2.2.13 How to obtain and parse coredump with ESP32? . . . 13
  - 2.2.14 How to do RF performance test with ESP32, ESP8266, and ESP32S2? . . . 13
  - 2.2.15 What are the reasons for not being able to recognize ESP devices under the Win10 system? 13
  - 2.2.16 One error occurred with ESP32 as `Core 1 panicked (Cache disabled but cache memory region accessed)`. What could be the reason? . . . 14
  - 2.2.17 How to read the flash model information of the modules? . . . 14
  - 2.2.18 When debugging the Ethernet Example in ESP-IDF, the following exception log appears. How can I resolve such issue? . . . 14

2.2.19 I found a "Brownout detector was triggered" failure on my ESP32. How to resolve such issue? . . . 14
2.2.20 After ESP32 imported the protocol_examples_common.h header file, the file cannot be found while compiling. What could be the reason? . . . 15
2.2.21 When using ESP8266 NonOS v3.0 SDK, the following error occurred. What could be the reason? . . . 15
2.2.22 When using flash_download_tool to flash firmware to the ESP8266 module, how to solve the following error? . . . 15
2.2.23 When upgrading from ESP-IDF v4.4 to v5.0 and above, the error *esp_log.h:265:27: error: format '%d' expects argument of type 'int', but argument 6 has type 'uint32_t' {aka 'long unsigned int'} [-Werror=format=]265 | #define LOG_COLOR(COLOR) "033[0;" COLOR "m"* is reported. How can this be resolved? . . . 15
2.2.24 Do ESP32 series products support the use of JTAG functionality in a boundary scan environment? Where can I download the BSDL files? . . . 15
2.3 Environment Setup . . . 16
2.3.1 When setting up ESP32-S2 environment using command `idf.py set-target esp32s2`, an error occurred as `Error: No such command 'set-target'`. What could be the reason? . . . 16
2.3.2 When installing ESP-IDF version master using ESP-IDF Tools 2.3 in Windows system, an error occurred as `Installation has failed with exit code 2`. What could be the reason? . . . 16
2.3.3 On Windows, when setting up the environment using esp-idf-tools, the following error occurs when running `make menuconfig`: . . . 16
2.3.4 During the installation process of esp-idf-tools on Windows, an exception occurs with the Python tool: . . . 16
2.3.5 What should I do if I get `Download failed: security channel support error` when installing build environment in the Windows system? . . . 17
2.3.6 When executing export.bat in Windows system, what should I do if I get CMake and gdbgui version errors? . . . 17
2.3.7 Errors occurred when using idf.menuconfig and idf.build after updating the ESP-IDF version from v3.3 to the latest one: . . . 17
2.3.8 How to configure `PATH` and `IDF_PATH` when developing ESP32 and ESP8266 simultaneously? . . . 17
2.3.9 Do I need to use command `idf.py set-target` every time when switching to another project? . . . 17
2.3.10 How to know the version of ESP-IDF, is it recorded in a certain document? . . . 18
2.3.11 How to optimize ESP-IDF compilation in Windows environment? . . . 18
2.3.12 Is there an esptool that can be used directly on Windows? . . . 18
2.3.13 What could be the cause of the error `KeyError: 'idfSelectedId'` when running `. /install.sh`? . . . 18
2.3.14 When running `demo`, the package manager component dependency cannot be pulled, with the failure `Invalid manifest format`, `Invalid dependency format`, and `unknown keys in dependency details: override_path`. What could be the reason? . . . 19
2.3.15 After installing the ESP-IDF CMD environment using the ESP-IDF v4.4.8-Offline Installer package, why does the following compilation error occur when directly compiling the hello_world example? . . . 19
2.3.16 How to fix the following error when building a project and pulling components? . . . 19
2.3.17 How to mirror esp-idf SDK and related toolchains from GitHub to domestic sources on Windows? . . . 20
2.3.18 Which operating systems and development environments/IDEs are supported by Espressif chips? . . . 20
2.4 Firmware Update . . . 21
2.4.1 How does the host MCU flash ESP32 via serial interfaces? . . . 21
2.4.2 How to download firmware for ESP32 series modules using the USB-Serial tool? . . . 21
2.4.3 How to flash firmware on macOS and Linux? . . . 21
2.4.4 Does ESP32 support programming using JTAG pins directly? . . . 21
2.4.5 Does ESP_Flash_Downloader_Tool support customized programming control? . . . 21

2.4.6 Can I enable the Secure Boot function for ESP32 via OTA? . . . 22
2.4.7 How to resolve the following error that occurred when flashing firmware to ESP32-S2 based on ESP-IDF v4.1? . . . 22
2.4.8 How to download firmware based on ESP-IDF using flash_download_tool? . . . 22
2.4.9 What is the communication protocol for flashing ESP chips? . . . 23
2.4.10 How to program ESP32-C3 ' s firmware offline? . . . 23
2.4.11 How to set the flash SPI mode to QIO mode on ESP32? . . . 23
2.4.12 After downloading the program and powering on EPS8266, the serial port printed the following log. What is the reason? . . . 23
2.4.13 What are the Espressif module programming tools? . . . 23
2.4.14 What is the difference between the Factory mode and Developer mode of the flash download tool? . . . 23
2.4.15 The ESP32-C3 chip should be able to conduct firmware download through USB, but I failed to do so under ESP-IDF v4.3. So, how can I use USB for firmware download? . . . 24
2.4.16 Why does the programming fail for the jig with a 4-port hub in factory mode? . . . 24
2.4.17 I ' m using an ESP32-WROVER-B module to download the AT firmware via the flash download tool. However, an error occurred after writing to flash. But the same operation succeeded when replacing the module with ESP32-WEOVER-E, what is the reason? . . . 24
2.4.18 Why can ' t the Flash Download Tools be used to reflash the firmware on a device that has enabled flash encryption but not disabled the download mode? . . . 24
2.4.19 When updating ESP32 firmware through the UART interface based on esptool serial port protocol, can I add a new app partition? . . . 25
2.4.20 I am using ESP8266 to download the firmware via the flash download tool. After downloading the firmware, there is no programming output log, and the serial port prints the following messages. What could be the reason? . . . 25
2.4.21 Why does my USB driver fail to be recognized by the Windows 7 system? . . . 25
2.4.22 After using the ESP32-WROVER-E module to download the program, the following log is printed after powered on. What could be the reason? . . . 25
2.4.23 When using the Flash Download Tools to flash ESP32-C3 via USB, 8-download data fail occurs repeatedly. How can I solve it? . . . 25
2.4.24 On ESP32, the bootloader.bin of ESP-IDF v3.0 can not start app.bin of ESP-IDF v5.0. Why? . . . 26
2.4.25 Does ESP32-C3 support disabling ROM code logs via OTA? . . . 26
2.4.26 Will the operation of other tasks be affected when the chip is undergoing an OTA firmware upgrade (esp_ota_write())? . . . 26
2.4.27 How many devices can be flashed at once in factory mode using Flash Download Tool? . . . 26
2.4.28 Does ESP32 OTA support resumable uploads? . . . 26
2.4.29 When using esptool for host computer development, after increasing the download baud rate to above 1,152,000, there is no significant improvement in download speed. Is this a limitation imposed by esptool? . . . 26

3 application solution 27
3.1 Artificial intelligence . . . 27
3.1.1 What types of cameras are supported on AI image recognition products? . . . 27
3.1.2 Which versions of ESP-IDF are supported by ESP-WHO? . . . 27
3.1.3 Is there any information about the WeChat mini program of ESP-EYE? . . . 27
3.1.4 What languages are supported by the esp-skainet demo? . . . 27
3.1.5 What model frameworks does ESP-DL support? . . . 28
3.1.6 Does ESP-DL support all models of the three platforms mentioned above (MXNet, Pytorch, and TensorFlow)? . . . 28
3.1.7 Can the model files of ESP-SKAINET be stored in the SD card? . . . 28
3.1.8 How to customize command words in ESP-SKAINET? . . . 28
3.1.9 How to reduce the system footprint of AI speech models? . . . 28
3.1.10 What is the difference between a 16-bit quantization model and an 8-bit quantization model? 28
3.1.11 How does the AI voice model modify the number of microphone channels? . . . 28
3.1.12 How do I get the actual audio captured in the development board? . . . 28
3.1.13 Do you have relevant study materials for ESP-SR GitHub? . . . 29
3.1.14 Do you have relevant study materials for ESP-DL? . . . 29

3.1.15 How does ESP32-S3 customize English command words for recognition? . . . 29
3.1.16 How to improve the detection efficiency of Yolo11? . . . 29
3.2 AT . . . 29
3.3 Audio Application Framework . . . 29
3.3.1 When using the VoIP feature of ESP-ADF, how to eliminate echo during calls between a mobile phone and an ESP32 device? . . . 30
3.3.2 Is there a reference example for accessing the big language model provided by Baidu? . . . 30
3.3.3 Does the official VoIP example provided by Espressif support RTP? . . . 30
3.3.4 Is the SIP protocol in ESP-ADF open-source? . . . 30
3.3.5 Can ESP-ADF examples implement the volume adjustment function of Bluetooth headsets? 30
3.3.6 I want to use a sensor on the I2C of ESP32-LyraT. Is there an example on how to read I2C device data? . . . 30
3.3.7 How to output 32-bit I2S audio data? . . . 30
3.3.8 Why do I always get an error when compiling example/get-started/play-mp3 with ESP-ADF and ESP-IDF v4.1? . . . 31
3.3.9 Where can I check the ESP-IDF version supported by ESP-ADF? . . . 31
3.3.10 Will integrating DuerOS occupy the recording function of the ESP32-LyraT development board throughout the process? . . . 31
3.3.11 Does ESP-ADF support customizing voice recognition wake-up words? . . . 31
3.3.12 Does ESP-ADF support running Alexa examples on the ESP32-LyraTD-MSC v2.1 development board? . . . 31
3.3.13 Regarding voice recognition on ESP32, can you recommend a suitable development board for localization? . . . 31
3.3.14 Does ESP32 have a development board that supports both MIC and AUX pickup? . . . 32
3.3.15 How to implement call function using the ESP32-LyraT development board? . . . 32
3.3.16 What is the maximum power of supported speakers for ESP32 series audio development board? . . . 32
3.3.17 Does Espressif's voice wake-up solution have specific requirements regarding environmental noise? . . . 32
3.3.18 If there is an AUX input on the ESP32's AI development board, can the MIC still be used to pick up sound? . . . 32
3.3.19 When designing an audio development board with the ESP32-WROVER-B module and ES8311, which pins can be selected for the MCLK clock? . . . 33
3.3.20 Can the ESP32-WROVER-E module use a single I2S to realize simultaneous playback and recording? . . . 33
3.3.21 Do Espressif modules support Spotify Connect? . . . 33
3.3.22 When running the korvo_du1906 example on an ESP32-Korvo-DU1906 board, a reboot caused the following error message: Guru Meditation Error: Core 0 panic'ed (IllegalInstruction). Exception was unhandled. How to resolve such issue? . . . 33
3.3.23 Can ESP-DSP fft run 4096, 8192, and more samples? . . . 33
3.3.24 How to connect a microphone to ESP32? . . . 33
3.3.25 Does ESP32 support analog audio or digital audio output? . . . 34
3.3.26 What audio formats does the ESP32 chip support? . . . 34
3.3.27 How to use the ESP32 chip to decode compressed audio? . . . 34
3.3.28 Where is the code example for ESP-LED-Strip? . . . 34
3.3.29 Does ESP32 support online voice recognition? . . . 34
3.3.30 Does ESP32 support volume adjustment of Bluetooth headphones? . . . 34
3.3.31 Can Wi-Fi and FFT be used simultaneously? . . . 34
3.3.32 What does the RECORD_HARDWARE_AEC macro mean when it is True or False in the ESP-ADF SDK? . . . 35
3.3.33 Are there any recommended product models and application references for developing an outdoor pure voice intercom solution based on Espressif's products? . . . 35
3.3.34 What is the difference between software AEC (Acoustic Echo Cancellation) and hardware AEC in the ESP-ADF audio application development framework? . . . 35
3.3.35 Do ESP series products support Voice Activity Detection (VAD)? . . . 35
3.4 ESP-BLE-MESH Development Framework . . . 35
3.4.1 Why does the ESP-BLE-MESH application wait for a long time during fast provisioning? . 35

3.4.2 How to clear the network information of the ESP32 ESP-BLE-MESH node? / How to manually reset the ESP-BLE-MESH device? . . . 36
3.4.3 How to delete the network information of a node from the provisioning device? . . . 36
3.4.4 When a node is powered off and then powered on again, do I need to re-provision it? . . . 36
3.4.5 Assuming development board 1 acts as a Provisioner, and boards 2, 3, and 4 act as nodes. After successful provisioning, if board 1 loses power and is then powered on again, can it join the mesh network? . . . 36
3.4.6 In ESP-BLE-MESH, how do I know if a node is offline? . . . 36
3.4.7 How do ESP-BLE-MESH nodes communicate with each other in strings? . . . 36
3.4.8 Failed in initializing the partition when configuring ESP-BLE-MESH to save node information. The error message is `BLE_MESH: Failed to init mesh partition, name ble_mesh, err 261`. How to fix it? . . . 37
3.4.9 How to add Health Model to the Provisioner's example? . . . 37
3.4.10 What's the difference between using ble_mesh_fast_prov_client as a Provisioner and a mobile phone as a Provisioner? . . . 37
3.4.11 Are there any tools or methods to view the encrypted messages transmitted between ESP-BLE-MESH nodes? . . . 37
3.4.12 Can the AppKey be set by the manufacturer? Is it related to the unicast address? . . . 37
3.4.13 If a node suddenly goes offline, does the entire mesh network need to poll and send Heartbeat messages through the Health Model monitoring mechanism? . . . 37
3.4.14 Can the main node (Proxy node) and the slave node send messages to each other using the client-server model? Is there an example? . . . 38
3.4.15 In the lower-right corner of the nRF mobile application, tap `Setting` and you will find a configurable `Network Key` field. Whose NetKey does this field configure? . . . 38
3.4.16 How can a device join the ESP-BLE-MESH network? . . . 38
3.4.17 What is the maximum packet size for ESP-BLE-MESH transmission? . . . 38
3.4.18 Can you provide an application example of provisioning through ESP32 ESP-BLE-MESH? What software can be used as the provisioning configuration application? . . . 38
3.4.19 In ESP-BLE-MESH, the default name of the unprovisioned device is ESP-BLE-MESH. How to modify it? . . . 38
3.4.20 How many node devices can the ESP32 ESP-BLE-MESH application connect to? . . . 38
3.4.21 After ESP32 runs the ESP-BLE-MESH program for a long time, a segmentation error occurs when the client sends a message to the server, and the ESP-BLE-MESH prints `NO multi-segment message contexts available`. How to solve the issue? . . . 39
3.4.22 Can I disable NetKey and IV Update when using the ESP32 ESP-BLE-MESH application? 39
3.4.23 How to receive other ADV broadcast packets while BLE MESH is enabled? . . . 39
3.5 Camera Application . . . 39
3.5.1 What type of camera does the ESP32 series chip support? . . . 39
3.5.2 What are the output image formats of the camera? . . . 39
3.5.3 What parameters does the camera support for adjustment? . . . 39
3.5.4 What is the relationship between MCLK and PCLK in the camera? What is the difference between them? . . . 40
3.5.5 Is a higher PCLK always better for the camera? . . . 40
3.5.6 Do the ESP32 series chips support the MIPI interface? . . . 40
3.5.7 Do the ESP32 series chips support the USB2.0 interface? . . . 40
3.5.8 Why is the transmission speed of YUV/RGB in the camera slower than JPEG? . . . 40
3.5.9 In camera applications, what are the factors that may affect the frame rate? . . . 41
3.5.10 How to troubleshoot when the camera fails to run? . . . 41
3.5.11 Does ESP32 support video stream transmission? . . . 41
3.5.12 Where is the factory firmware of ESP-EYE? . . . 41
3.5.13 Where are the examples related to the camera solution stored? . . . 42
3.5.14 Does ESP32 support a camera with a 12-bit DVP interface? . . . 42
3.5.15 Can ESP32 use a camera without JEPG encoding to obtain JPEG images? . . . 42
3.5.16 Can the 2-megapixel OV2640 camera on ESP-EYE be changed to only output 300,000-pixel images? . . . 42
3.5.17 Does ESP32 support a global shutter camera? . . . 42
3.5.18 What is the frame rate when ESP32 transfers 1080P video via RTSP using the DVP camera? 42

3.5.19 ESP32-S3 only supports MJPEG encoding, but H264/H265 format encoding is needed when implementing rtsp/rtmp streaming. Is there any encoding that supports H264/H265 format? . . . 42
3.5.20 Does ESP32/ESP32-S3 support wide-angle cameras? . . . 43
3.5.21 It takes five seconds for ESP32-S2 to display the camera image from power-on. Is there room for improvement? . . . 43
3.5.22 Can ESP32 directly support 24 MHz frequency to the GC0308 camera? . . . 43
3.5.23 Does ESP32/ESP32-S3 support the MMS streaming protocol? . . . 43
3.5.24 When debugging the GC2145 camera with ESP32-S3, the maximum supported resolution seems to be 1024x768. If it is adjusted to a larger resolution, such as 1280x720, it will print cam_hal: EV-EOF-OVF error. How to solve this issue? . . . 43
3.5.25 Does ESP32-S3 support the GB28181 protocol? . . . 43
3.5.26 Is there any reference for ESP32/ESP32-S2/ESP32-S3 to recognize the QR code through the camera? . . . 44
3.5.27 When adding the SD-card interface and camera interface for OV5640 sensor, we found that some pins of different ESP32 drivers conflicted with each other. Please suggest pins for the camera interface and SD-card interface. . . . 44
3.5.28 Can a driver for a specific camera model be added if the currently supported camera sensors do not meet my requirements? . . . 44
3.5.29 How to add a custom resolution? . . . 44
3.5.30 How to modify the register configuration of the camera sensor? . . . 44
3.5.31 What is the reason for triggering "cam_hal: EV-VSYNC-OVF" in esp32-camera? . . . 45
3.5.32 What could be the reason for the following warning log appearing in the Camera application based on ESP32-S3? . . . 45
3.5.33 What is the difference between the two capture modes of the ESP32-Camera? . . . 45
3.5.34 How to implement frame skipping with the esp32-camera SDK? . . . 46
3.5.35 Can ESP32-S3 connect to two cameras and display split screen? . . . 46
3.5.36 Does ESP32-S3 support a 10-bit DVP camera? . . . 46
3.5.37 How to scale image data? . . . 46
3.5.38 What's the difference between the two camera application components esp32-camera and esp-video? . . . 46
3.5.39 How to add a new camera driver for applications using the esp-video framework? . . . 46
3.5.40 Can I connect multiple cameras to ESP32-P4? . . . 46
3.5.41 Is it necessary to pair a camera sensor that can only output RAW format data with an ISP module to output clear images? . . . 46
3.5.42 What equipment is needed to debug the ISP module? . . . 47
3.5.43 How does esp-video-components output XVCLK to the camera sensor? . . . 47
3.6 ESP Matter . . . 47
3.6.1 Which ESP modules can be connected to Matter? . . . 47
3.6.2 What learning materials are available to get started with ESP Matter? . . . 47
3.6.3 Can I compile and develop ESP Matter on Windows? . . . 47
3.6.4 Where can I find the Matter protocol documentation? . . . 47
3.6.5 How to register an ESP Matter product? . . . 48
3.6.6 When using Ubuntu virtual machine to develop ESP Matter on ESP32-C3, network provisioning failed when I followed the Matter official tutorial. What could be the reason? . . . 48
3.6.7 How to apply for a Matter DAC? . . . 48
3.6.8 Does ESP Matter have a test tool/test app? . . . 48
3.6.9 Matter needs to use DCL in the network provision process. What is the specific function of DCL? . . . 48
3.6.10 How to connect our Zigbee-based products with Matter through ESP chip? . . . 48
3.6.11 Does Matter work with Samsung's smartthings? . . . 49
3.6.12 Can Matter-enabled ESP devices be remotely controlled using Amazon/Google/Apple voice devices? Do these voice devices need to support the Matter protocol? (For example: Saying "Turn off the light" to turn off the lights in the house) . . . 49
3.6.13 Does the product need to pass WiFi authentication and Bluetooth BQB authentication before submitting the Matter authentication? . . . 49
3.6.14 Where is the DAC (Device Attestation Certificate) pre-imported by ESP Matter module stored? . . . 49

3.6.15 Can I configure Wi-Fi of ESP32 Matter devices by BLE? . . . 49
3.6.16 How to solve the following error when provisioning the device with the command `pairing ble-wifi` or `pairing code-wifi` using the light example in esp-matter via chiptool? . . . 50
3.6.17 How to generate and install Matter QR codes during mass production? . . . 50
3.7 ESP-NOW . . . 50
3.7.1 What is the one-to-one bit rate for ESP32 in ESP-NOW mode? . . . 50
3.7.2 What is ESP-NOW? What are its advantages and application scenarios? . . . 50
3.7.3 Can Wi-Fi be used with ESP-NOW at the same time? . . . 51
3.7.4 How do I set the rate at which ESP-NOW data is sent? . . . 51
3.7.5 ESP-NOW allows pairing with a maximum of 20 devices. Is there a way to control more devices? . . . 51
3.7.6 What is the maximum number of devices that can be controlled by ESP-NOW? . . . 51
3.7.7 Do I need to connect a router for communication between ESP-NOW devices? . . . 51
3.7.8 Why does ESP-NOW limit the data length of each packet to 250 bytes? Can it be configured? 51
3.7.9 What should I pay attention to when using ESP-NOW applications? . . . 52
3.7.10 How can I reduce power consumption when using ESP-NOW? . . . 52
3.7.11 In addition to wireless communication through ESP-NOW, is there any other better way to realize one-to-one and one-to-many communication? . . . 52
3.7.12 Do ESP-NOW applications support sending packets over each Wi-Fi channel? . . . 52
3.7.13 Are there any special procedures required if I intend to use ESP-NOW for commercial purposes? Could you provide technical documentation on ESP-NOW? To evaluate the quality of wireless communication, I would like to know relevant information about parameters including CSMA/CA, modulation method, and bit rate. . . . 52
3.7.14 I tested the application esp-idf/examples/wifi/espnow using ESP32. Does it only support connecting to 7 encrypted devices at the maximum? . . . 53
3.7.15 How do I obtain the corresponding RSSI when transferring data using ESP-NOW? . . . 53
3.7.16 How do I use RSSI in ESP-NOW to achieve selective range control? . . . 53
3.8 ESP RainMaker cloud service . . . 53
3.9 Third-Party Cloud Service . . . 53
3.9.1 Are there any examples for OTA upgrading? . . . 53
3.9.2 How to integrate ESP32 with Tmall Genie? Any reference materials? . . . 53
3.9.3 What is the difference between `esp-aliyun` and `esp-ali-smartliving`? . . . 54
3.9.4 How to ask questions and provide feedback when encountering issues in connecting Espressif products to the Espressif cloud platform? . . . 54
3.9.5 Can ESP32 and ESP8266 connect to Alexa or Google Home? . . . 54
3.9.6 What should I do to connect ESP32 to Alibaba Cloud via Ethernet and MQTT? . . . 54
3.9.7 How to generate triplet bin files in bulk on Alibaba Cloud? . . . 54
3.10 ESP-WIFI-MESH Application Framework . . . 54
3.10.1 How much memory does ESP-WIFI-MESH occupy? Is external PSRAM required? . . . 55
3.10.2 Can ESP-WIFI-MESH support performing OTA for multiple devices at a time? . . . 55
3.10.3 How many devices can ESP32 support for ESP-WIFI-MESH networking? . . . 55
3.10.4 What is the difference between the ESP-WIFI-MESH Router mode and the No Router mode of ESP32? . . . 55
3.10.5 Can ESP-WIFI-MESH of ESP32 complete networking when the child node fails to detect the router signal? . . . 55
3.10.6 Can ESP-WIFI-MESH of ESP32 automatically repair the network? . . . 55
3.10.7 When using ESP-WIFI-MESH of ESP32 without connecting to Wi-Fi, how can I realize a self-organizing network? . . . 55
3.10.8 When using ESP-WIFI-MESH of ESP32 for automatic root node election during networking, is it possible to designate specific local modules for the election process? . . . 56
3.10.9 Can multiple root nodes send messages to each other in a no-router scenario using ESP-WIFI-MESH of ESP32? . . . 56
3.10.10 How to query the source code of the ESP-WIFI-MESH application? . . . 56
3.10.11 Does ESP-WIFI-MESH offer a no-router solution for self-networking? . . . 56
3.10.12 After using Mwifi for automatic networking, how can I obtain the signal strength (RSSI) of all potential parent nodes of a specific node? . . . 56
3.10.13 What protocol is used for communication between nodes in the ESP-MDF Mesh network? 56

3.10.14 Can ESP-WIFI-MESH connect all nodes to the router? . . . 56
3.10.15 Can the root node of ESP-WIFI-MESH connect to the Internet via 4G dial-up? . . . 57
3.10.16 After a successful connection using the esp_mesh_set_parent function, if the AP is disconnected, the function will continuously attempt to reconnect. How can I set the number of reconnect attempts? . . . 57
3.10.17 Why is `phy_init: failed to load RF calibration data` reported after setting the button? . . . 57
3.10.18 How to pause/resume Mwifi? . . . 57
3.10.19 For the ESP32 series without router Mesh networking, how can an application connect to the softAP of the root interface? . . . 57
3.10.20 ESP-WIFI-MESH can connect to an AP but cannot connect to the TCP server on that AP. How can this issue be resolved? . . . 57
3.10.21 How to modify the AP connection and maximum layer in the Mwifi example? What is the maximum bandwidth and delay during communication? . . . 58
3.10.22 How to get real-time return values of the sensor? . . . 58
3.10.23 For a new node that might already be installed in a device that is located at a considerable distance from the root node, how can this node join the ESP-WIFI-MESH network? . . . 58
3.10.24 Is the source code for the ESP-WIFI-MESH application open? . . . 58
3.10.25 What is the maximum packet size for Wi-Fi Mesh data transmission? . . . 59
3.10.26 Does ESP32's Wi-Fi Mesh support self-networking without a router? . . . 59
3.10.27 What is the maximum number of node layers allowed when using ESP32 with Wi-Fi Mesh? 59
3.10.28 When testing the esp-mdf/examples/function_demo/mwifi/router example with the ESP32 development board, the device name displayed on the router connection end is "espressif" after the ESP32 connects to the router. How can I change this name? . . . 59
3.10.29 Can Wi-Fi Mesh send messages to specific nodes through a TCP server? . . . 59
3.10.30 During the operation of the ESP32 Wi-Fi Mesh network, what event does the system feedback if the root node is lost? . . . 59
3.10.31 When developing Wi-Fi Mesh applications with ESP32 using the esp_mesh_send() function, I've observed that the server is not receiving any data. How can data be transmitted from leaf nodes to an external server? . . . 59
3.10.32 How to perform an OTA upgrade after ESP-MESH devices are networked? . . . 60
3.10.33 Is there a reference design for ESP-MESH lights? . . . 60
3.10.34 What is the default mode of ESP-MESH nodes? . . . 60
3.10.35 When ESP-MESH starts in AP+STA mode, can the phone search for the AP? . . . 60
3.10.36 After the device has completed networking, is it necessary to perform a complete re-scan when adding a new device? . . . 60
3.10.37 Can ESP32, as the master device, achieve time synchronization with multiple slave devices with an error within 2 ms? . . . 60
3.10.38 How to get the node type in ESP-MESH? . . . 60
3.10.39 Is there an example of ESP-Mesh root node sending messages to the server via Ethernet? . 61
3.10.40 Does the ESP-Mesh-Lite solution support applications without a router? . . . 61
3.10.41 When ESP-WIFI-MESH is networked, is it possible for the root node or child nodes to simultaneously initiate a Wi-Fi Scan to scan for available AP information around? . . . 61
3.10.42 When using the ESP-WIFI-MESH Router solution, how to switch to a new router for networking? . . . 61
3.10.43 Does the ESP-MESH-LITE application support Wi-Fi Power Save mode? . . . 61
3.10.44 Both devices are root nodes connected to the router but the mesh network is not formed. How to fix this? . . . 62
3.10.45 When using the Mesh-Lite App for provisioning, the device is connected to Wi-Fi 1 and then switched to Wi-Fi 2, but after `esp_restart` it still connects to Wi-Fi 1. How to fix this? . . . 62
3.10.46 Can Mesh-Lite child nodes directly maintain an MQTT long connection? . . . 62

4 Software framework **63**
4.1 Bluetooth . . . 63
4.1.1 Bluetooth LE & Bluetooth . . . 63
4.1.2 Classic Bluetooth . . . 78
4.1.3 Other Bluetooth . . . 80

4.2 Ethernet . . . 81
4.2.1 When building an example on ESP32 Ethernet development board, an error occurred as "emac: Reset EMAC Timeout" . What could be the reasons? . . . 81
4.2.2 When ESP32 connected to LAN8720 externally, with GPIO0 providing CLK, the initialization of Ethernet example failed. How to resolve such issue? . . . 81
4.2.3 When using the Ethernet example in ESP-IDF, an error occurred as "Timed out waiting for PHY register 0x3 to have value 0xc0f0 (mask 0xfff0). Current value 0xffff" . How to resolve such issue? . . . 82
4.2.4 When using ESP-IDF v4.1, how to set the static IP for ESP32 Ethernet? . . . 82
4.2.5 Is there any impact on Ethernet functionality if replacing the module of ESP32-Ethernet-Kit with ESP32-WROOM-32D? . . . 83
4.2.6 When using ESP32 to design a self-developed Ethernet board, after downloaded the official esp-idf/examples/ethernet example, errors are reported as follows, what is the reason? . . . 83
4.2.7 Does ESP32 Ethernet support MII interface? . . . 83
4.2.8 Is it possible to connect ESP32-S2 to Ethernet externally? . . . 84
4.2.9 Do the ESP32 series chips support to use EMAC and SPI-Ethernet modules simultaneously? 84
4.2.10 How to diagnose packet delay issues in ESP32 Fast Ethernet transmission? . . . 84
4.3 coexistence . . . 84
4.3.1 When Wi-Fi coexists with Bluetooth, what mode does it support? . . . 84
4.3.2 When Wi-Fi coexists with ESP-BLE-MESH, the Wi-Fi throughput is low, why? . . . 84
4.3.3 Does ESP32 support coexistence between ESP-WIFI-MESH and Bluetooth® LE Mesh? . 85
4.3.4 Does ESP32 support coexistence between Bluetooth® and Wi-Fi? . . . 85
4.3.5 When Wi-Fi, Bluetooth® LE, and A2DP sink coexist, audio data reception is lost and lagged while entering Bluetooth LE scanning. How to resolve such issue? . . . 85
4.3.6 Can BLE advertising (Connectable) and iBeacon sending (advertising) coexist? . . . 85
4.3.7 How do ESP32 Bluetooth® and Bluetooth® LE dual-mode coexist and how can I use this coexistence mode? . . . 85
4.3.8 How do ESP32 Bluetooth® and Wi-Fi coexist? . . . 86
4.3.9 How can I resolve the frequently occurred ELxXX error (such as ELx200) when Wi-Fi and Ble co-exit? . . . 86
4.4 Peripherals . . . 86
4.4.1 Analog-to-Digital Converter (ADC) . . . 86
4.4.2 Digital-to-Analog Converter (DAC) . . . 90
4.4.3 GPIO & RTC GPIO . . . 90
4.4.4 Inter-Integrated Circuit (I2C) . . . 94
4.4.5 Inter-IC Sound (I2S) . . . 96
4.4.6 LCD . . . 97
4.4.7 LED Control (LEDC) . . . 106
4.4.8 Motor Control Pulse Width Modulator (MCPWM) . . . 108
4.4.9 Pulse Counter (PCNT) . . . 108
4.4.10 Remote Control Transceiver (RMT) . . . 109
4.4.11 Secure Digital Input Output (SDIO) . . . 111
4.4.12 Serial Peripheral Interface (SPI) . . . 113
4.4.13 Timer . . . 117
4.4.14 Touch Sensor . . . 117
4.4.15 Two-Wire Automotive Interface (TWAI) . . . 119
4.4.16 Universal Asynchronous Receiver/Transmitter (UART) . . . 119
4.4.17 USB . . . 122
4.4.18 Other Peripherals . . . 131
4.5 Protocols . . . 132
4.5.1 ESP-TLS . . . 132
4.5.2 HTTP . . . 135
4.5.3 lwIP . . . 140
4.5.4 Mbed TLS . . . 150
4.5.5 MQTT . . . 153
4.5.6 Other Protocols . . . 158
4.6 Provisioning . . . 162

4.6.1 Can I add any broadcast data I want to Android ESP-Touch (e.g., add a device ID so that ESP32 can receive this ID)? . . . 162
4.7 Security . . . 162
4.7.1 Is the firmware in ESP8266 readable? . . . 162
4.7.2 Is it possible to encrypt firmware for ESP8285? . . . 162
4.7.3 What is the difference between secure boot v1 and v2? . . . 162
4.7.4 After enabling secure boot, there is a build error indicating missing files. What could be the reasons? . . . 163
4.7.5 After enabling secure boot, is it possible for modules to be flashed again? . . . 163
4.7.6 With flash encryption enabled, a module reports an error as `flash read error` after reflashed. How to resolve such issue? . . . 163
4.7.7 After enabling flash encryption and secure boot for ESP32, how to disable them? . . . 163
4.7.8 Is there any security strategy for ESP32 to protect its firmware? . . . 163
4.7.9 When ESP32 debugging GDB after enabling flash encryption, why does it continuously reset and restart? . . . 164
4.7.10 How to enable flash encryption for ESP32? . . . 164
4.7.11 After GPIO0 is pulled down, the ESP32 cannot enter download mode and prints "download mode is disable". What could be the reason? . . . 164
4.7.12 Can the secure boot function be enabled for ESP32 in Arduino development environment? 164
4.7.13 What are the use scenarios for secure boot and flash encryption? . . . 164
4.7.14 What are the data stored in eFuse involved in secure boot and flash encryption? . . . 164
4.7.15 Enabling secure boot failed with the log "Checksum failure". How to fix it? . . . 165
4.7.16 NVS encryption failed to start and an error occurred as `nvs: Failed to read NVS security cfg: [0x1117] (ESP_ERR_NVS_CORRUPT_KEY_PART)`. How can I solve this issue? . . . 165
4.7.17 After flash encryption was enabled, a warning occurred as `esp_image: image at 0x520000 has invalid magic byte (nothing flashed here)`. How can I solve this issue? . . . 165
4.7.18 Why is reltead data not encrypted after I enable `CONFIG_EFUSE_VIRTUAL` and flash encryption? . . . 165
4.7.19 Can I update an app firmware which enables flash encryption in a device which does not enable fash encryption through OTA? . . . 165
4.7.20 How can I delete keys of secure boot? . . . 165
4.7.21 After I enabled secure boot or flash encryption (development mode), I cannot flash the new firmware, and an error occurred as `Failed to enter Flash download mode`. How can I solve this issue? . . . 165
4.7.22 After I input the command `espefuse.py read_protect_efuse BLOCK3 command` in the terminal configured with ESP-IDF to enable the read-protection for Efuse BLOCK3, why is the data of the Efuse BLOCK3 all 0x00 when I input `esp_efuse_read_block()` to read the Efuse BLOCK3? . . . 166
4.7.23 How can I enable secure boot or flash encryption by pre-burning eFuse? . . . 166
4.7.24 After enabling Secure Boot, why can't the new bootloader.bin be flashed using the `idf.py flash` command? . . . 166
4.7.25 After enabling Secure Boot or flash encryption, how can I view the security-related information in the device? . . . 166
4.7.26 After enabling Secure Boot or flash encryption, what should I pay attention to during OTA (Over-The-Air) updates? . . . 166
4.7.27 Which USB functions will be disabled after the ESP32-S3 enables flash encryption or Secure Boot? . . . 167
4.7.28 After enabling flash encryption, if there are multiple flash encryption keys for `XTS_AES_128_KEY` in the device's eFuse, how will the device select the key? . . . 167
4.7.29 When Secure Boot V2 is enabled, how can I store the public key used for signature verification on the device? . . . 167
4.7.30 After enabling the Secure Boot V2 feature on ESP series products, is it still possible to reflash the firmware? . . . 167
4.7.31 Does the flash encryption scheme support encryption for the file system? . . . 167
4.7.32 How to create an encrypted `nvs_data.bin` when using NVS encryption? . . . 168
4.7.33 Does the SPIFFS file system based on ESP32 support the flash encryption scheme? . . . 168

4.7.34 Using the ESP32-C3 SDK based on ESP-IDF v5.0.6, NVS encryption based on Flash Encryption is enabled in the software configuration. After the device completes Flash Encryption and restarts the firmware, the firmware runs with the following error. What is the reason? . . . 168
4.7.35 How to implement firmware signature verification (ECDSA) at the application level? . . . 169
4.7.36 How to perform encryption on custom data? . . . 169
4.7.37 What is the difference between the NVS encryption scheme based on the Flash encryption framework and the NVS encryption scheme based on HMAC? . . . 169
4.8 Storage . . . 169
4.8.1 FAT Filesystem . . . 169
4.8.2 Non-Volatile Storage (NVS) . . . 171
4.8.3 PSRAM . . . 173
4.8.4 SD/SDIO/MMC Driver . . . 175
4.8.5 SPI Flash . . . 176
4.8.6 SPIFFS Filesystem . . . 178
4.8.7 Other Storages . . . 179
4.9 System . . . 180
4.9.1 How do I disable the watchdog if my application doesn't need it? . . . 180
4.9.2 What are the differences between RTOS SDK and Non-OS SDK? . . . 180
4.9.3 Why does the log output ets_main.c when ESP8266 starts? . . . 181
4.9.4 Why do I get compile errors when using IRAM_ATTR in Non-OS SDK? . . . 181
4.9.5 Where is the main function of ESP8266? . . . 181
4.9.6 Which part of ESP8266 partition-tables should be paid special attention to? . . . 181
4.9.7 Is it possible to compile the binaries in application layer and bottom layer separately? . . . 181
4.9.8 What should I pay attention to when using flash of ESP32 module at 80 MHz? . . . 181
4.9.9 How can I download programs into a single-core module using ESP-IDF? . . . 182
4.9.10 How to enable the dual-core mode of ESP32 with ESP-IDF? . . . 182
4.9.11 Can the ESP32-D0WD chip store user programs? . . . 182
4.9.12 Will the data in PSRAM be lost when ESP32 enters low power mode? . . . 182
4.9.13 Is the ESP32 CPU system time generated by the system tick clock? What is the precision? 182
4.9.14 How can I modify the clock frequency of ESP32's flash and PSRAM? . . . 182
4.9.15 When using ESP32-SOLO-1 modules, what settings should I make to run ESP-IDF on a single-core module? . . . 183
4.9.16 Can time_t be configured to be 64 bits in ESP-IDF? (It is currently 32 bits) . . . 183
4.9.17 How does the firmware distinguish whether the main chip is ESP8285 or ESP8266? . . . 183
4.9.18 Can ESP32 load library files as dynamic libraries? . . . 183
4.9.19 How can I reduce the IRAM occupied by the ESP32 system? . . . 183
4.9.20 What is the low voltage reset threshold of the ESP32 chip? . . . 184
4.9.21 Why is ESP32 automatically woken up in the light sleep example? . . . 184
4.9.22 When testing ESP32 deep_sleep example, why does the program crash into a dead loop when const int wakeup_time_sec = 3600? . . . 184
4.9.23 How many system reset methods does ESP32 have? . . . 184
4.9.24 Why does ESP8266-NONOS-V3.0 SDK output the following error? . . . 184
4.9.25 Can ESP32 use the entire 8 MB of PSRAM memory? . . . 185
4.9.26 After ESP8266 AT connects to AP, the system enters into Modem-sleep by default, but the current does not drop significantly. What are the reasons? . . . 185
4.9.27 Can ESP32 permanently change the MAC address? . . . 185
4.9.28 How can ESP8266 verify all.bin as an illegal file during OTA upgrade? . . . 185
4.9.29 Where are the release notes after the ESP-IDF version is updated? . . . 186
4.9.30 Does ESP8266 have a detailed register manual? . . . 186
4.9.31 ESP32 cannot start normally after enabling Secure Boot and outputs the following error. What is the reason for it? . . . 186
4.9.32 How does ESP8266 save data during a software reboot? . . . 186
4.9.33 What timers are available on ESP8266? . . . 186
4.9.34 What is the purpose of the watchdog on ESP8266? . . . 187
4.9.35 What should be paid attention to when using `user_init` in ESP8266? . . . 187

4.9.36 Why does the system keep restarting after ESP32 enables both `Enable debug tracing of PM using GPIOs` and `Allow .bss segment placed in external memory`? . . . 187
4.9.37 Why does the program trigger RTCWDT_RTC_RESET when ESP32 bootloader in ESP-IDF v3.3 version runs app bin in v3.1 version? . . . 187
4.9.38 Does the ESP32 chip come with a unique chip_id from the factory? . . . 188
4.9.39 How to check ESP8266 reset cause? . . . 188
4.9.40 How can I optimize the size of binary files compiled by ESP32? . . . 188
4.9.41 Does ESP32 have an API for rebooting the system? . . . 188
4.9.42 What is the reason for the exception log `invalid header: 0xffffffff` of ESP32? 188
4.9.43 What is the timed mechanism to wake up ESP8266 from deep sleep? . . . 188
4.9.44 Why is the ESP32 RAM obtained using `heap_caps_get_free_size` about 300 KB, instead of the 520 KB provided in the manual? . . . 189
4.9.45 How to perform OTA upgrade via LAN app for ESP32 & ESP8266? . . . 189
4.9.46 How can I modify the GPIO used by the serial port for log output in ESP32? . . . 189
4.9.47 ESP8266 uses MQTT ssl_mutual_auth to communicate. Why does the following error occur during OTA? . . . 189
4.9.48 There are NVS options in ESP32's `menuconfig > Component config`. Why are the configuration items are empty? . . . 189
4.9.49 Does ESP32 undergo a random watchdog reset after power-up or waking up from Deep-sleep? 189
4.9.50 When the ESP32 CPU uses cache to access external SRAM, if these operations need to be processed by the CPU at the same time, will read and write errors occur? . . . 190
4.9.51 ESP32 CPU may freeze when it switches directly from 240 MHz to 80/160 MHz. How can I solve it? . . . 190
4.9.52 For ESP32 pads with both GPIO and RTC_GPIO functions, the pull-up and pull-down resistors can only be controlled by the RTC_GPIO pull-up and pull-down registers. How can I solve it? . . . 190
4.9.53 As the flash startup speed is slower than that of the chip reading flash, ESP32 may randomly reset the watchdog once after power-on or waking up from Deep-sleep. How can I solve it? 190
4.9.54 When the ESP32 CPU accesses the external SRAM, there is a small probability of causing reading and writing errors. How can I solve it? . . . 190
4.9.55 For a dual-core ESP32 chip, when one CPU's bus is reading address space A, and the other CPU's bus is reading address space B, the CPU reading address space B may encounter an error. How can I solve it? . . . 191
4.9.56 The ESP32 CPU resets the interrupt signal of the CAN controller by reading the `INTERRUPT_REG` register. If the CAN controller happens to generate a transmission interrupt signal within the same APB clock cycle, the transmission interrupt signal is lost. How can I solve it? . . . 191
4.9.57 When the program meets the following conditions simultaneously, ESP32 v3.0 will be live-locked, causing the CPU to remain in a memory access state and unable to continue executing instructions. How can this be resolved? . . . 191
4.9.58 The ESP32 CPU has restrictions when accessing the address spaces `0x3FF0_0000 ~ 0x3FF1_EFFF` and `0x3FF4_0000 ~ 0x3FF7_FFFF`. How can this be resolved? . . . 191
4.9.59 How can I turn off log output in ESP32? . . . 192
4.9.60 Can the data stored in RTC Memory run when ESP8266 is in Deep-sleep mode? . . . 192
4.9.61 What is the maximum length of the NVS Key for ESP32? . . . 192
4.9.62 Does cJSON in ESP-IDF release/v4.2 support uint64_t data parsing? . . . 192
4.9.63 Given that the GDB debugging function is working before the flash encryption is disabled, then why does the device keep restarting during the GDB debugging after the flash encryption is enabled? . . . 192
4.9.64 When ESP32 uses mobile hotspot for OTA firmware download, if the data switch is turned off for a few seconds and then turned on again, the program will always be stuck in OTA (the same applies when unplugging and plugging the WAN cable when using a router). Why? 193
4.9.65 Which GPIOs can be used to wake up ESP32-C3 in Deep-sleep mode? . . . 193
4.9.66 When using the ESP-WROOM-02D module with a battery for power supply, are there any risks in frequently formatted reading and writing flash as the battery is low (the module barely starts up)? . . . 193
4.9.67 How can I check the maximum stack size used by a thread for ESP32? . . . 193

4.9.68 Why does ESP32 print the "SW_CPU_RESET" log? . . . 193
4.9.69 For ESP32 products, when testing NVS separately, I found it occupies a lot of memory. What is the reason? . . . 194
4.9.70 How do I change the system time of a module? . . . 194
4.9.71 During the OTA upgrade process, an ESP_ERR_OTA_VALIDATE_FAILED error occurred after calling esp_ota_end. How can I troubleshoot such issue? . . . 194
4.9.72 How does ESP8266-RTOS-SDK store data to RTC memory? . . . 194
4.9.73 After waking up from Deep-sleep mode, where does ESP8266 boot? . . . 194
4.9.74 When will the RTC clock be reset? . . . 195
4.9.75 Can ESP32 be woken up by pulling EN low after entering Deep-sleep mode with the AT+GSLP command? . . . 195
4.9.76 When multiple threads want to use the watchdog of ESP32, should each thread enable the watchdog individually? . . . 195
4.9.77 How to enter Light-sleep mode with ESP8266-RTOS-SDK release/v3.3? . . . 195
4.9.78 How can I wake up ESP8266 from Deep-sleep mode? . . . 195
4.9.79 When using the ESP32-WROVER* module, there is a problem of battery jitter or abnormal power-off and power-on, causing the system to crash and fail to wake up. What is the reason? 195
4.9.80 How can I flash a customized MAC address? . . . 196
4.9.81 When ESP32 uses esp_timer, network communication or Bluetooth communication is abnormal. What is the reason? . . . 196
4.9.82 With ESP32, are there any return instructions if I skip to a function using the jump instruction in ULP? . . . 196
4.9.83 How to adjust the warning level for project build? . . . 196
4.9.84 The firmware compiled based on the ESP-IDF SDK varies as it contains the information about IDF_PATH and compilation time. How to remove that information? . . . 197
4.9.85 When I downloaded the official application hello_world using ESP32-S3-DevKitM-1, the following error occurred. What is the reason for that? . . . 197
4.9.86 What is the accuracy of the internal 150 kHz RTC of ESP32 series chips? . . . 197
4.9.87 What versions of ESP-IDF SDK are supported by ESP32-D0WDR2-V3 chip? . . . 197
4.9.88 When testing OTA applications based on the ESP32 chip, can I delete the default factory partition in the partition table and set the address of the OTA_0 partition to 0x10000? . . . 197
4.9.89 Why can't the espefuse.py burn_key command be used to burn ESP32-C3 eFuse BLOCK3? . . . 198
4.9.90 Why I encountered the following error after I run the firmware based on the ESP-IDF SDK? 198
4.9.91 How to solve the issue of being unable to parse due to spaces in the specified url during Wi-Fi OTA? . . . 198
4.9.92 How to get the version number of newlib of ESP-IDF? . . . 198
4.9.93 Does ESP32-P4 support floating-point arithmetic? . . . 198
4.9.94 What are the ESP-IDF versions supported by different series of ESP32 chip versions? . . . 198
4.9.95 How to check the descriptions of error codes defined in ESP-IDF? . . . 199
4.9.96 Why does the following error occur during firmware operation? What could be the common reason? . . . 199
4.9.97 Does the ESP chip support using the esp-bootloader-plus compression upgrade scheme to upgrade other MCUs? . . . 199
4.9.98 When developing applications based on ESP32, how can I obtain information such as task status, task priority, remaining task stack, and the core used by the task? . . . 199
4.9.99 How to obtain the CPU usage of a task when developing applications with ESP32? . . . 199
4.9.100 After downloading the firmware, ESP32 fails to start normally, and the log shows the following information. What could be the reason? . . . 199
4.9.101 Does it support redirecting the UART0 output logs of ESP32 to the file system? . . . 200
4.9.102 Can the ESP32 BootLoader, which is configured to run in single-core mode, be upgraded to dual-core mode via OTA? . . . 200
4.9.103 Does the ESP32 series chip support OTA firmware updates via File Transfer Protocol (FTP)? 200
4.9.104 After using the native_ota_example for an OTA upgrade, why does the device enter the ESP_OTA_IMG_UNDEFINED event, with the ota_state value printed as -1, even though the firmware works normally? . . . 200
4.9.105 What are the differences between sdkconfig, sdkconfig.default, and sdkconfig.old in an ESP-IDF project? . . . 200

4.9.106 What could possibly be the reason for the "Not digestSign key in json doc of OTA" error that occurs during an OTA upgrade? . . . 201
4.9.107 Are there any firmware update solutions that can help reduce flash memory and transmission traffic? . . . 201
4.9.108 Can the crystal oscillator glitch detector be disabled via software? . . . 201
4.9.109 Do ESP32-C2 and ESP32-C3 built-in flash support the suspend function? After enabling suspend in the configuration, the log shows `Flash suspend feature is enabled`. What should I do? . . . 201
4.9.110 Is UART wakeup reliable in Light Sleep mode? . . . 202
4.9.111 Does the ESP32 series of chips support decompression? . . . 202
4.9.112 Does ESP-IDF v5.5.3 support ESP32-P4 v1.3? . . . 202
4.10 Thread . . . 203
4.10.1 Does ESP32-H2, as an OpenThread RCP device, support building a Thread border router based on a Linux host? . . . 203
4.10.2 Does the ESP chip equipped with an IEEE 802.15.4 module and supporting Thread allow interaction in the form of AT-like commands? . . . 203
4.11 Wi-Fi . . . 203
4.11.1 Do ESP32 and ESP8266 support Chinese SSID for Wi-Fi? . . . 203
4.11.2 How much time does an ESP32 scan take? . . . 204
4.11.3 [Scan] Do Espressif's products support boundary scans? . . . 204
4.11.4 What is the definition for Wi-Fi channel? Can I select any channel of my choice? . . . 204
4.11.5 [LWIP] With ESP-IDF v4.1, how to configure ESP32's IP address when it is in SoftAP mode? . . . 204
4.11.6 [LWIP] How to configure ESP32's static IP when it is in Station mode? . . . 205
4.11.7 [LWIP] How to configure the Option contents of DHCP Server in ESP-IDF? . . . 205
4.11.8 [Performance] How to test the bit rate of Wi-Fi modules? . . . 206
4.11.9 [LWIP] What is the default IP address of ESP8266 SoftAP? . . . 206
4.11.10 Why do I have problem connecting to router with IP 192.168.4.X in SoftAP + Station mode? 206
4.11.11 [Connect] How many devices is ESP8266 able to connect in SoftAP mode? . . . 206
4.11.12 Do ESP8266/ESP32/ESP32-S2/S3/C2/C3 support web/SoftAP provisioning? . . . 207
4.11.13 [Connect] How do ESP8266 and ESP32 hide SSID in SoftAP mode? . . . 207
4.11.14 Does the buffer parameter in esp_wifi_802.11_tx interface include FCS? . . . 207
4.11.15 What is the supported Wi-Fi frequency band and power meter for ESP32? . . . 207
4.11.16 What is the maximum value of ESP32 Wi-Fi RF power? . . . 207
4.11.17 How does ESP32 adjust Wi-Fi TX power? . . . 208
4.11.18 [Connect] How many devices is ESP32 able to connect in AP mode? . . . 208
4.11.19 [Connect] How do Wi-Fi modules rank signal strength levels based on RSSI values? . . . 208
4.11.20 [Connect] Why does ESP32 disconnect from STA when it is in Soft-AP mode? . . . 208
4.11.21 [Connect] While ESP32 connecting Wi-Fi, how can I determine the reason of failure by error codes? . . . 209
4.11.22 Does ESP32 perform domain name resolution each time it connects to the server? . . . 209
4.11.23 [Connect] What does the number after the state machine switch in Wi-Fi log mean? . . . 209
4.11.24 [Connect] What does bcn_timeout, ap_probe_send_start mean? . . . 210
4.11.25 [Connect] How to reconnect Wi-Fi after it disconnected? . . . 210
4.11.26 [Connect] When does ESP32 disconnect from SoftAP in station mode? . . . 210
4.11.27 [Scan] Why does the STA cannot find any AP sometimes during the scanning? . . . 210
4.11.28 [Scan] What is the maximum number of APs that can be scanned? . . . 211
4.11.29 [Scan] Can I choose to connect the best AP when there are multiple APs with identical ssid/password during the scan? . . . 211
4.11.30 [Scan] How to configure scan_method in the wifi_sta_config_t structure? What is the difference between all_channel_scan and fast_scan? . . . 211
4.11.31 [LWIP] How to get error code of the socket? . . . 211
4.11.32 [LWIP] What is the default keep-alive time of TCP? . . . 211
4.11.33 [LWIP] What is the retransmission interval of TCP? . . . 212
4.11.34 [LWIP] What is the maximum number of sockets that can be created? . . . 212
4.11.35 [Sleep] What kinds of sleeping mode does ESP32 have? What are the differences? . . . 212
4.11.36 [Sleep] How can I enable the dynamic frequency scaling function for ESP32 in Modem-sleep mode? . . . 212

4.11.37 [Sleep] How low can the speedstep function go for ESP32 in modem sleep mode? 212
4.11.38 [Sleep] What affects the average current of ESP32 in modem sleep mode? 213
4.11.39 [Sleep] Why the average current measured in modem sleep mode is a bit high? 213
4.11.40 [Sleep] Why the average current measured in light sleep mode is a bit high? 213
4.11.41 [Sleep] What kinds of power-saving modes does ESP32 have? What are the differences? . 213
4.11.42 Does ESP8266 support 802.11k/v/r protocol? 214
4.11.43 Does ESP32 Wi-Fi support roaming between different APs with the same SSID? 214
4.11.44 [Connect] After the NONOS_SDK updated from version *2.1.0* to *2.2.2*, why does the connecting time become longer? 214
4.11.45 How does ESP32 receive and transmit Wi-Fi 802.11 packets? 214
4.11.46 [Connect] The ESP32 and ESP8266 failed to connect to router, what could be the reasons? 214
4.11.47 [Connect] What kind of networking methods does ESP8266 have? 214
4.11.48 [Connect] What are the specifications of Wi-Fi parameters when using SmartConfig? . . . 215
4.11.49 [Connect] Does ESP8266 Wi-Fi support WPA2 enterprise-level encryption? 215
4.11.50 [Connect] What are the low-power modes for ESP32 to maintain its connection to Wi-Fi? 215
4.11.51 Do Espressif's chips support WPA3? 215
4.11.52 [Connect] How does the device choose AP when there are multiple identical SSIDs in the current environment? 215
4.11.53 [Connect] Does ESP8266 have repeater solutions? 215
4.11.54 What is the retransmission time for ESP32's data frame and management frame? Can this be configured? 216
4.11.55 How to customize the hostname for ESP32? 216
4.11.56 How to obtain 802.11 Wi-Fi packets? 216
4.11.57 Does ESP32 Wi-Fi support PMF (Protected Management Frames) and PFS (Perfect Forward Secrecy)? 216
4.11.58 Why does ESP8266 print out an AES PN error log when using esptouch v2? 216
4.11.59 Does ESP32 WFA certification support multicast? 216
4.11.60 When using ESP32 to establish a hotspot, can I scan all APs and the occupied channels first, and then select the smallest and cleanest channel to establish my own AP? 216
4.11.61 I'm scanning Wi-Fi on an ESP32 device using release/v3.3 version of ESP-IDF. When there are some identical SSIDs, same SSID names will show in the Wi-Fi list repeatedly. Is there an API to filter such repeated names? 217
4.11.62 Does ESP8266 support EDCF (AC) scheme? 217
4.11.63 I'm using the master version of ESP8266-RTOS-SDK to open the WiFi Qos application to get EDCF support. How does ESP8266 decide which data packet should be allocated to the EDCF AC category? 217
4.11.64 Using ESP32, how to configure the maximum Wi-Fi transmission speed and stability without considering memory and power consumption? 217
4.11.65 In Wi-Fi SoftAP mode, how many Station devices can ESP8266 be connected at most? . . 217
4.11.66 How to get CSI data when using ESP32 device in Station mode? 217
4.11.67 In AP + STA mode, after an ESP32 is connected to Wi-Fi, will the Wi-Fi connection be affected if I enable or disable its AP mode? 217
4.11.68 I'm using ESP-IDF release/v3.3 for ESP32 development, but only bluetooth function is needed, how to disable Wi-Fi function through software? 218
4.11.69 In ESP-IDF, the `esp_wifi_80211_tx()` interface can only be used to send data packets, is there a corresponding function to receive packets? 218
4.11.70 What are the reasons for the high failure rate of esptouch networking? 218
4.11.71 How to optimize the IRAM when ESP32 uses Wi-Fi? 218
4.11.72 How to test ESP32's Wi-Fi transmission distance? 219
4.11.73 What is the maximum length of Wi-Fi MTU for an ESP32 and how to change it? 219
4.11.74 During the on-hook test for an ESP32 device, the following log shows. What does it mean? 219
4.11.75 How to disable AP mode when ESP32 is in AP + STA mode? 219
4.11.76 After ESP32 used the Wi-Fi function, are all ADC2 channels unavailable? 219
4.11.77 How do I set the country code for a Wi-Fi module ? 220
4.11.78 When using ESP32 as a SoftAP and have it connected to an iPhone, a warning prompts as "low security WPA/WPA2(TKIP) is not secure. If this is your wireless LAN, please configure the router to use WPA2(AES) or WPA3 security type", how to solve it? 220

4.11.79 Since ESP32's Wi-Fi module only supports 2.4 GHz of bandwidth, can Wi-Fi networking succeed when using a multi-frequency router with both 2.4 GHz and 5 GHz of bandwidth? 220
4.11.80 How to obtain the RSSI of the station connected when ESP32 is used in AP mode? 220
4.11.81 Does ESP32 support FTM(Fine Timing Measurement)? 221
4.11.82 When ESP32 is in STA+AP mode, how to specify whether using STA or AP interface to send data? 221
4.11.83 ESP8266 wpa2_enterprise How to enable Wi-Fi debugging function? 221
4.11.84 Is there a standard for the number of Wi-Fi signal frames? 222
4.11.85 What is the current progress of WFA bugs fixing? 222
4.11.86 When Wi-Fi connection failed, what does the error code mean? 222
4.11.87 When using ESP32's Release/v3.3 of SDK to download the Station example, the device cannot be connected to an unencrypted Wi-Fi. What is the reason? 222
4.11.88 What is the maximum PHY rate of Wi-Fi communication of ESP32-S2 chip? 222
4.11.89 Does ESP modules support EAP-FAST? 223
4.11.90 Does ESP modules support the WiFi NAN (Neighbor Awareness Networking) protocol? . 223
4.11.91 When using ESP32 with release/v3.3 version of ESP-IDF. When configuring the router, is there an API to directly tell that the entered password is wrong? 223
4.11.92 When testing the Station example of ESP32 base on v4.4 version of ESP-IDF, how to support WPA3 encryption mode? 223
4.11.93 How does ESP32 speed up the Wi-Fi connection? 223
4.11.94 Does ESP32 WPA2 Enterprise Authentication support Cisco CCKM mode? 224
4.11.95 Using wpa2_enterprise (EAP-TLS method), what is the maximum length supported for client certificates? 224
4.11.96 Does ESP8089 support Wi-Fi Direct mode? 224
4.11.97 How does ESP32 connect to an AP whose RSSI does not fall below the configured threshold when there are multiple APs in the environment? 224
4.11.98 ESP32 Wi-Fi has a beacon lost and sends 5 probe requests to the AP after 6 seconds. If the AP does not respond, disconnection will be caused. Can this 6 seconds be configured? 224
4.11.99 Does ESP32 Wi-Fi work with PSRAM? 224
4.11.100[Connect] How to troubleshoot the issue that ESP32 series of products cannot connect to the router over Wi-Fi from the hardware and software aspects? 224
4.11.101After being connected to the router, ESP32 prints `W (798209) wifi:<ba-add>idx:0 (ifx:0, f0:2f:74:9b:20:78), tid:0, ssn:154, winSize:64` and `W (798216) wifi:<ba-del>idx` several times every 5 minutes and consumes much more power. Why? 225
4.11.102Why can't ESP32 keep the Wi-Fi sending rate at a fixed value with the function esp_wifi_config_80211_tx_rate() to maintain stable transmission? 225
4.11.103How do I debug the ESP32 station that is connected to a router but does not get an IP properly? 225
4.11.104When ESP32 works as a softAP, the station connected to it does not get the IP. How to debug? 226
4.11.105In ESP-IDF menuconfig, after `Component config > PHY > Max Wi-Fi TX power(dBm)` is configured to adjust the Wi-Fi transmit power, what is the actual power? For example, what is the actual maximum transmit power when the option is configured to 17 dBm? 226
4.11.106ESP-IDF currently supports connecting to Chinese SSID routers with UTF-8 encoding. Is there a way to connect to Chinese SSID routers with GB2312 encoding? 226
4.11.107After connecting to the router, ESP32 consumes much power in an idle state, with an average current of about 60 mA. How to troubleshoot the issue? 226
4.11.108How to configure the Wi-Fi country code when the ESP end product needs to be sold worldwide? 227
4.11.109Sometimes the rate drops or even a disconnection occurs after a period of iperf testing. What is the reason and how to solve it? 227
4.11.110Why is this log frequently printed when the phone connects to the ESP32-S3 that works as the Wi-Fi AP based on the ESP-IDF v5.0 SDK? 228
4.11.111Does ESP32 support the coexistence of the network port (LAN8720) and Wi-Fi (Wifi-AP)? 228
4.11.112How can I optimize ESP32's slow IP address acquisition after Wi-Fi is connected in a weak network environment or interference environment? 228

4.11.113When ESP32/ESP32-S2/ESP32-S3 series chips work in SoftAP mode, they are susceptible to disconnect from mobile phones and PCs of other manufacturers when they are communicating with each other. How can I optimize this situation? 228
4.11.114What is the value range of ESP32 Wi-Fi TX power? 228
4.11.115How do I get Wi-Fi RSSI when using ESP32? 229
4.11.116Does ESP32 support WPA3-Enterprise? 229
4.11.117Does ESP modules support WAPI (Wireless LAN Authentication and Privacy Infrastructure)? 229
4.11.118How can I increase the time for scanning routers when using ESP32 as the Wi-Fi Station to connect routers? 229
4.11.119Does ESP32 support LDPC? 230
4.11.120Does the ESP module support WAPI AS? 230
4.11.121When encountering the error message `Error occurred during sending: errno 12` during UDP transmission testing, what should I do to resolve it? 230
4.11.122Does the ESP module support Wi-Fi HaLow functionality? 230
4.11.123How to prevent any radio frequency waves from being sent in ESP32 Wi-Fi Scan mode? . 230
4.11.124What is the default 802.11 Wi-Fi protocol when the ESP32-C6 is in Wi-Fi AP mode? . . 230
4.11.125Why can't the ESP32 Wi-Fi Station connect to a Wi-Fi hotspot in 2.4 GHz Enhanced Open mode? 231
4.11.126How to determine whether the connected Wi-Fi applies Wi-Fi 4 or Wi-Fi 6? 231
4.11.127Does ESP32-S3 support AP and STA working simultaneously? 231
4.11.128Does the ESP Wi-Fi module support power save mode in SoftAp mode? 231
4.11.129What is the maximum ranging bandwidth supported by ESP32S3 FTM? 231
4.11.130Do ESP chips support one STA conducting FTM with multiple APs at the same time? . . 231
4.11.131In WiFi Station mode, which WiFi mode will ESP32-C6 use to establish a connection with the router when both WiFi 4 and WiFi 6 modes are enabled on the router? 231
4.11.132In WiFi AP mode, how many WiFi Station devices can ESP32-C2 series products support at most? 232
4.11.133Does ESP32 support WPA3 WiFi AP mode? 232
4.11.134Do ESP chips support Short GI in SoftAP mode? 232
4.11.135Do STA and AP need to establish a connection first before executing FTM? 232
4.11.136Do ESP chips support uAPSD in SoftAP mode? 232
4.11.137When enabling WiFi AP mode on products based on Espressif series, is it possible to set up automatic channels? 232
4.11.138Do ESP series products support roaming in Wi-Fi AP mode? 232
4.11.139Does the router's DHCP lease duration affect the module's Wi-Fi connection? 233
4.11.140If the module's authentication mode is set to WIFI_AUTH_OPEN, but the router uses a higher-security authentication mode like WIFI_AUTH_WPA3_PSK, can the module still connect? 233
4.11.141How to handle the issue "esp-aes: Failed to allocate memory"? 233
4.11.142How to completely disable Wi-Fi configurations on ESP32 to optimize firmware size when Wi-Fi features are not required? 233
4.11.143Why can't a router with WPA2 encryption be connected when the WiFi scan auth mode threshold is set to WPA2_WPA3_PSK in menuconfig, but only routers with WPA3 or WPA2/WPA3 encryption can? 233
4.11.144What is the coexistence relationship between ESP32-C6's Wi-Fi 6 mode and HT40? . . 233
4.11.145When running the ESP8266_RTOS_SDK/examples/wifi/getting_started/station example on ESP8266, it cannot connect to the esp-idf/examples/wifi/getting_started/softAP example on ESP32-S3 with esp-idf v5.3 as Wi-Fi AP mode hotspot. However, it can connect to the esp-idf/examples/wifi/getting_started/softAP example on ESP32-S3 with esp-idf v5.0 as Wi-Fi AP mode hotspot. What is the reason? 234
4.11.146Does ESP32-C5 support simultaneous use of 2.4GHz and 5GHz Wi-Fi? 234
4.11.147When using an ESP32 device, can I enable smartconfig provisioning (esp_smartconfig_start) while maintaining an existing Wi-Fi connection, so as to switch to another Wi-Fi at any time? 234
4.11.148After successful Wi-Fi provisioning with the Blufi example, why does the chip automatically reconnect to the previously configured Wi-Fi hotspot after reset and restart? 234
4.11.149Why doesn't the AP receive an ACK when the Wi-Fi STA sends data? 235

4.11.150 Why does the power consumption increase when `wifi:bcn_timeout, ap_probe_send_start` appears in the Wi-Fi STA log? . . . 235
4.11.151 Can ESP32-C5 sniff Wi-Fi 5 and Wi-Fi 6 packets at 20 MHz? . . . 235
4.11.152 Why does the ESP32 timeout when pinging a public IP through a 4G dial-up connection? . 235
4.11.153 Within a local area network, is it possible to find the MAC address of a mobile phone through UDP data content without using ARP? . . . 235
4.11.154 Does the Wi-Fi of ESP32-C5 support the simultaneous use of sniffer (promiscuous mode) and STA mode? . . . 235
4.11.155 When enabling the 5G band softAP on C5, will DFS be activated to avoid interference with radar on the same channel? . . . 236
4.11.156 How to configure the working frequency band (2.4 GHz/5.8 GHz) of ESP32-C5? . . . 236
4.11.157 Does Espressif chips support the ARP aging mechanism? What is the aging time, and can it be modified? For example, when connecting via Wi-Fi, the MAC address of the AP is obtained through ARP. How long is the interval before the next ARP request is sent? . . . 236
4.11.158 What does `LWIP_GARP_TMR_INTERVAL` do? What is the interval in seconds for sending ARP? . . . 236
4.11.159 After disabling modem sleep, the ping latency is still high. How can it be further optimized? 236
4.11.160 When using a packet capture tool to analyze Wi-Fi traffic, is it normal that the data remains encrypted even after entering the SSID and password? . . . 237
4.11.161 Does ESP32-C5 support AP mode on the 5GHz band? Do 2.4 GHz and 5 GHz share the same MAC address in AP mode? . . . 237
4.12 Zigbee . . . 237
4.12.1 Does the Zigbee protocol stack of ESP32-H2 and ESP32-C6 support automatic frequency hopping under interference? . . . 237

5 Hardware related **239**
5.1 Chip Comparison . . . 239
5.1.1 What is the difference between ESP32 with a single core and ESP32 with dual cores regarding programming method, feature performance, power consumption, and so on? . . . 239
5.1.2 What's the difference between ESP32 v3.0 and previous chip revisions in software and hardware? . . . 239
5.1.3 Can the GPIO34 ~ GPIO39 of ESP32 only be set to input mode? . . . 240
5.1.4 Does ESP32 support a driver for Linux? . . . 240
5.1.5 What is the meaning of the acquired data when you scan the data matrix on the module's shielding case? . . . 240
5.1.6 Can VDD3P3_RTC of ESP32 support independent battery power supply? . . . 240
5.1.7 What is the difference between ESP32-PICO-D4, ESP32-PICO-V3, and ESP32-PICO-V3-02? . . . 240
5.1.8 Does ESP8266 support 32 MHz crystal? . . . 241
5.1.9 Do ESP products support Zephyr? . . . 241
5.1.10 How to identify the chip revision from the chip silk marking? . . . 241
5.2 Development board . . . 241
5.2.1 Does the ESP32-Korvo v1.1 development board integrate an LED driver chip? . . . 241
5.2.2 How to improve the overheating situation of the ESP-EYE development board? . . . 242
5.2.3 If a development board is not powered up through USB, how to use pins to supply power for it? . . . 242
5.2.4 In ESP32-Korvo-DU1906, how does the DU1906 chip interact with ESP32 for audio data? 242
5.2.5 Is there an Ethernet development board that supports POE power supply? . . . 242
5.2.6 The LED light on the ESP32-DevKitC development board does not light up and the device manager cannot find the device. Why? . . . 242
5.2.7 Why can't I find the EN button mentioned in the documentation on the development board? 242
5.2.8 I can't find the serial port in the device manager after connecting the ESP32 development board to a Windows computer. What could be the reasons? . . . 243
5.2.9 For the ESP32-LyraT v4.3 audio development board, it's hard to enter download mode even when the Boot button is held down for a long time. What could be the reason? . . . 243
5.2.10 How long does it take for the ESP-WROOM-02D module to restart after the reset signal? . 243
5.2.11 According to the schematic of ESP32-LyraT-Mini, the analog output of the ES8311 codec chip is connected to the input of the ES7243 ADC chip. What is the purpose of this? . . . 243

5.2.12 When using the ESP32-MINI-1 module, the serial port printed the follows log when powered on. What could be the reason? . . . 243
5.2.13 Which GPIO is connected to the RGB LED of the ESP32-S3-DevKitC-1 development board? . . . 244
5.3 Hardware Design . . . 244
5.3.1 The I2S pins of ESP32 are scattered. Can I route I2S signals to adjacent pins? For example, to GPIO5, GPIO18, GPIO23, GPIO19, and GPIO22, or to GPIO25, GPIO26, GPIO32, and GPIO33. . . . 244
5.3.2 How can I stop the power loss of VDD3P3_RTC after ESP32 enters Light-sleep mode? . 244
5.3.3 What should be noted when I configure the pins of ESP32? . . . 245
5.3.4 What is the voltage tolerance of GPIOs of ESP chips? . . . 245
5.3.5 What are the power supply specifications for ESP8266? . . . 245
5.3.6 Do Espressif Wi-Fi modules support single-layer PCBs? . . . 246
5.3.7 What should be noted when I power ESP8266 with batteries? . . . 246
5.3.8 Where can I get the footprint of ESP32 Series? . . . 246
5.3.9 For ESP32-S2 chips, can I have audio connection when the DVP camera interface is in use? 246
5.3.10 What should be noted when I assign I2C signals to GPIO0 and GPIO4 of ESP32 modules? 246
5.3.11 When the external flash is connected to GPIO6 ~ GPIO11, can they be set as SPI pins? . . 246
5.3.12 Do I need to connect an external crystal when using the ESP8285 chip? . . . 246
5.3.13 Where can I find the reference design for connecting an external PSRAM to ESP32-D2WD? 247
5.3.14 Can I use ESP32 to play music with PWM or DAC? . . . 247
5.3.15 Why is the suggested voltage range of ESP32 modules different from that of ESP32 chips? 247
5.3.16 Why does it take a longer time to erase the flash of self-developed modules than that of Espressif modules? . . . 247
5.3.17 Why does the current surge when ESP8266 is powered on? . . . 247
5.3.18 What choices do I have when configuring the RMII clock for the Ethernet of ESP32? . . . 248
5.3.19 What kind of socket is used on ESP32-LyraT development boards to connect a speaker? . 248
5.3.20 For modules housing ESP32, which pins cannot be set by users? . . . 248
5.3.21 Which is the reset pin of ESP32? . . . 248
5.3.22 What should be noted when I design the power supply for ESP8266? . . . 248
5.3.23 When I use the TOUT pin of ESP8266 to collect ADC sample signals, will the pins be damaged if the voltage is greater than 1.0 V? . . . 248
5.3.24 For modules with PCB antennas, what should be noted when I design the PCB and the housing of the antenna? . . . 249
5.3.25 Can GPIO 34 ~ GPIO39 of ESP32 be used as UART RX pins? . . . 249
5.3.26 Where can I find the design reference for the external 32 kHz crystal of ESP32 modules? . 249
5.3.27 Does the flash of ESP32 modules support 80 MHz QIO access mode? . . . 249
5.3.28 How to configure the RMII synchronous clock for the Ethernet of ESP32? . . . 249
5.3.29 How can I hard reset ESP8266? Is hard reset active low or active high? What are the requirements for reset? . . . 249
5.3.30 What does the term NC mean in Espressif schematics? . . . 250
5.3.31 How can I use multiple antennas with ESP32-S2? . . . 250
5.3.32 Does ESP32-C3F SPI CS0 pin need an external 10 kΩ pull-up resistor? . . . 250
5.3.33 Is there any hardware design reference for ESP-Skainet Speech Recognition? . . . 250
5.3.34 Is it necessary to connect a 32 kHz RTC crystal to the ESP32 chip in hardware design? . . 251
5.3.35 For the ESP32-MINI-1 module, is there a component library for Altium Designer? . . . 251
5.3.36 Can I change the input voltage of UART0 of ESP8266 from 3.3 V to 1.8 V? . . . 251
5.3.37 Is the level of UART0 of ESP8266 determined by VDD (VCC_WIFI) or VDDPST (VCC_CODEC_IO)? . . . 251
5.3.38 What should be noted when I connect an external PSRAM to ESP32-D2WD? . . . 251
5.3.39 When the VDD power supply of ESP32 slowly rises from 0 V to 3.3 V, why does the chip not start as usual? . . . 252
5.3.40 When using the ESP32-WROOM-32D module, can I set GPIO12 for other uses? . . . 252
5.3.41 When connecting an external flash to ESP32-WROOM-32D module, is it possible if I do not use GPIO6 ~ GPIO11 pins? . . . 252
5.3.42 Do I need to add a shielding case to the PCB of ESP32 modules? . . . 252
5.3.43 Do I must use GPIO0, GPIO1, or GPIO3 of ESP32 as the I2S CLK pin? . . . 253
5.3.44 Does the ESP32-U4WDH chip support external PSRAM chips? . . . 253

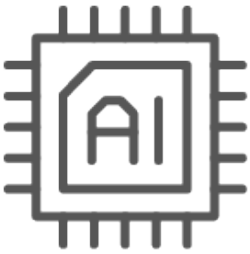
5.3.45 Does ESP32 support connection to an external SD NAND flash chip (instead of the default NOR flash chip) via the SPI0/SPI1 interface for storing application firmware? 253
5.3.46 Does it support to connect a second PSRAM chip externally based on the ESP32-S3R8 chip? 253
5.3.47 Could you please provide the 3D model and Footprint files of the ESP32-S3-WROOM-1 module? 253
5.3.48 Does ESP32/ESP32-S2/ESP32-C3/ESP32-S3 support powering the RTC power domain only to keep the chip working with low power consumption? 253
5.3.49 How can I improve the EMC performance? 254
5.3.50 Why do I need to connect a 499 Ω resistor to U0TXD for ESP32-S3? 254
5.3.51 How to calibrate the ESP32-S3 ADC in hardware? 254
5.3.52 How to design an automatic download circuit based on the ESP32 series chip? 254
5.3.53 Which crystal oscillator should be used on the ESP8266 chip? 254
5.3.54 Do the ESP32-C2, ESP32-C3 and ESP32-C6 chips support external PSRAM chips? . . . 254
5.3.55 When the ESP32-C3 is powered by a battery, it may fail to start if the supply voltage gradually decreases, for example, when the battery is fully discharged and then recharged. In such cases, the solution could only be disconnecting the battery from the ESP32-C3 and reconnect a fully charged battery, or to connect a voltage regulator diode between the 3.3 V pin and the EN pin to ensure the chip starts properly. What is the root cause of this situation? Is there an optimal solution? 255
5.3.56 Does the 3x3 GND grid on the ESP32 series module need to be copper-plated? 255
5.3.57 Why does ESP32 fail to start upon first power-up, and it requires a reset to start normally? 255
5.3.58 What could be the possible hardware-level reasons for the failure of Ethernet PHY initialization? 255
5.3.59 Ethernet PHY initialization is successful, but unable to obtain an IP, what could be the possible reasons at the hardware level? 255
5.3.60 What could be the reason that a PCB designed with the ESP32 chip can not connect to the router properly? 255
5.3.61 Is it possible to change the default power-up reset initial state of GPIO6 (JTAG pin MTCK) of ESP32-C6, which is input enabled and internal weak pull-up resistor enabled (IE & WPU), to input enable only (IE) by burning eFuse? 256
5.3.62 Can the VBAT pin of ESP32-H2 be powered independently? 256
5.3.63 Is ESP32-P4 VBAT (Backup Battery) now fully supported? 256
5.4 RF Related 256
5.4.1 Does the RF performance of an ESP32 module degrade if it runs with a 2.8 V supply? . . 256
5.4.2 What are the modulation methods supported by ESP chips? 256
5.4.3 How can I get the RF related information (e.g., antenna specification, antenna pattern, etc.) for certification? 257
5.4.4 Why does ESP32 automatically reduce its transmit power when it uses the RF Test Tool at 80 °C? 257
5.4.5 How to improve the receiving distance and strength of Wi-Fi signals for ESP32-WROVER-E? (Application scenario: Wi-Fi probe) 257
5.4.6 How to write phy_init data to flash ? 257
5.4.7 How can I optimize the second harmonic and other spurious signals created by my own products? 258
5.4.8 How can I suppress the harmonics of 80 MHz? 258
5.4.9 Is manual power calibration required for the Espressif modules that connect to external antennas? 258
5.4.10 What is the default duty cycle for the "default" level when using the "RF Test Tool" to set the Wi-Fi "TX continues" mode? 258
5.4.11 Does the ESP RF Test Tool support single carrier testing? 258
5.4.12 Is there a corresponding software API for the ESP RF Test Tool to implement the relevant RF test functions in the application firmware? 259
5.4.13 Is it feasible to design a 50-ohm impedance on an FPC? 259
5.4.14 Can the Espressif U series antenna modules with IPEX antenna connectors (such as ESP32-C3-WROOM-02U) be directly connected to SMA antennas? 259
5.4.15 Can I reuse Espressif's FCC/CE certification reports for the ESP32-S3-WROOM-1 module without additional fixed-frequency testing? 259

5.4.16 What could be the reason for the signaling test results being worse when provisioning with Bluetooth first, then testing Wi-Fi after turning off Bluetooth, compared to testing Wi-Fi alone? . . . 259
5.5 Process and ESD Protection . . . 259
5.5.1 What should be paid attention to during the ESP32 ESD test? . . . 260
5.6 Production Test . . . 260
5.6.1 Why can some modules download firmware normally when using DIO/DOUT, but encounter program abnormality when using QOUT/QIO? . . . 260
5.6.2 How to get the production test tool? . . . 260
5.6.3 When I use the `esptool.py burn_custom_mac` command to write the user-defined MAC address, why is the MAC address read by the `esptool.py read_mac` command still factory default? . . . 260
5.6.4 When downloading bin files to ESP32-WROVER-E (16 MB flash) using Flash Download Tool, multiple separate bin files can be downloaded successfully, but downloading the combined firmware (12 MB) failed. Why? . . . 261
5.6.5 Where can I find the MTBF report for a chip or module? . . . 261

6 Test verification **263**
6.1 Power Consumption Verification . . . 263
6.1.1 Why does ESP32 reboot when it is woken up from Deep-sleep mode? . . . 263
6.1.2 What sleep modes does ESP32 support? What is the difference between them? . . . 263
6.1.3 Can ESP32 in Deep-sleep mode be woken up by any RTC_GPIO? . . . 264
6.1.4 What is the power consumption of ESP8266 when the CHIP_PU pin is at the low level? . . . 264
6.1.5 Why does the minimum current of ESP32 in Light-sleep increase when the timer is not used as a wakeup source? . . . 264
6.1.6 In ESP32's Deep-sleep mode, using an internal 150 KHz RTC clock or using an external 32 KHz, which consumes more power? . . . 264
6.1.7 What are the requirements for CPU frequency to ensure normal operation of the RF module when reducing power consumption by reducing the CPU frequency? . . . 264
6.1.8 When I run the light-sleep example on ESP32-S3 modules, if I only use the GPIO wake-up source and do not enable the timer to wake up, the power consumption is 3 mA, which is significantly different from that on the datasheet. Why? . . . 264
6.1.9 When using Timer for wake-up based on the esp-idf/examples/system/deep_sleep example, although the wake-up time is set to 2.5 hours, it wakes up around 1 hour. What could be the reason? . . . 265
6.1.10 After enabling the Ethernet function based on the ESP32-WROVER-E module, the chip cannot enter the Power Save (Auto Light-sleep) mode. What is the reason? . . . 265

7 Commercial FAQ **267**
7.1 Which certificates have your products passed? . . . 267
7.2 Does your company have the ISO Quality Management System Certification? . . . 267
7.3 Do your chips and modules have environmental certificates such as REACH, ROHS, etc? . . . 267
7.4 Do you have distributors in China, Europe, the United States and Canada? . . . 267
7.5 How can I start a distribution business with Espressif? . . . 268
7.6 Where can I find your product information? Which of your products are in mass production? . . . 268
7.7 Do your products have a longevity commitment? . . . 268
7.8 Where can I find the SPQ (Standard Pack Quantity) and MOQ (Minimum Order Quantity) for your products? . . . 268
7.9 What is your recommended purchasing method? . . . 268
7.10 What's the price for bulk purchasing? How can I purchase in bulk? . . . 268
7.11 Where can I find all the differences between your products (e.g. in terms of series and types)? . . . 268
7.12 Do your products have firmware? Can I customize my module/chip flash before the product leaves the factory? How much does this process cost? How long does it take? How can you help me do this? 269
7.13 Which of your products support HomeKit? Where can I get the Espressif HomeKit SDK? . . . 269
7.14 What is your company's address? . . . 269
7.15 How can I contact your technical team? . . . 269
7.16 How can I get in touch with your company? . . . 269
7.17 How can I tell if an Espressif module is in mass production or an NPI product? . . . 270

ESP-FAQ is a summary document of frequently asked questions released by Espressif. This repository aims to help our users to quickly locate those questions and get answers through simple explanations. The current categories of FAQ cover: development environment, application solution, software framework, hardware related and test verification.

Instruction	Development environment	Application solution
		
Software framework	Hardware related	Test verification

Chapter 1

Instruction

This section provides guidance on how to use ESP-FAQ. The "Question search" guides you through the repository's search methods and categories, helping you find information more quickly. You are also welcome to contribute directly to ESP-FAQ by fixing bugs or adding new documents. For more details on how to contribute, please refer to "Document contribution".

1.1 Question search

This instruction includes the following two parts:

- Question search techniques
- Question classification framework

1.1.1 Question Search Techniques

Currently, there are mainly two searching techniques:

- Search keywords
- Exclude a specific keyword

Search key words

Extract keywords from your question and search for them, then the search results should list the best matches.

For example, if you expect to ask a question as `What is the Bluetooth LE Throughput for ESP32?`

Then just searching keywords such as `ESP32`, `BLE` and `throughput` should give you the result.

Exclude a Specific Keyword

Add a tag − into the search content in the format: `keyword -excluded keyword`. By doing so, the search results will not show the specific keyword you excluded.

For example, if you search `ESP32 -ble`, then any results with `ble` inside will not be shown.

1.1.2 Question Classification Categories

Once you have mastered the above mentioned `question search techniques`, you can use the categories in ESP-FAQ for reference to extract keywords for the questions you expect to ask and then search for them. The framework of ESP-FAQ categories is shown as follows:

- *Development environment*
- *Application solution*
- *Software framework*
- *Hardware related*
- *Test verification*
- *Commercial FAQ*

1.2 Document contribution

We welcome all the contributions to esp-faq project, such as fixing bugs, adding new documents and etc. We will accept new requests via Github Pull Requests.

1.2.1 Commit process

This section provides a brief overview of the `Add new items` and `Modify contents` processes. For the specific requirements during the processes, please refer to the links provided.

Here, we do not provide further operational instructions on `git`, please see Git learning material.

Add new items

1. *Create a new branch* following the *Branch naming conventions*;
2. Find the corresponding `*.rst` file locally or on web IDE, then add new items according to the template formats;
3. After your writing finished, you can check the document in the preview interface and build it using *Local build environment* to see if it has any formatting issues;
4. Push your branch to github and commit a pull request following the *Commit message standards*;
5. If all the abovementioned steps are finished following requirements, then you can *Submit a merge request*;
6. After all the review comments resolved and new pull requests updated, then this process is fully completed.

Modify contents

1. *Create a new branch* locally following the *Branch naming conventions*;
2. Find the corresponding `*.rst` file locally or on web IDE, then modify the desired contents;
3. After the modification finished, you can check the document in the preview interface and build it using the *Local build environment* to see if it has any formatting issues;
4. Push your branch to github and commit a pull request following the *Commit message standards*;
5. If all the abovementioned steps are finished following requirements, then you can *Submit a merge request*;
6. After all the review comments resolved and new pull requests updated, this process is fully completed.

1.2.2 Create a new branch

All the new branches are based on the **master branch**, so please make sure your current branch is the one you expect to merge.

For example:

```
git status #To see the status of your current branch
git checkout -b add/artificial-intelligence_camera_model #To add new contents␣
↪about "artificial-intelligence camera model"
```

1.2.3 Branch naming conventions

- Add a new item: `add/artificial-intelligence_{q&a}`, `{q&a}` is the brief English name of the file. For example, if you expect to add a new item as `artificial intelligence camera model`, then the branch name should be: `add/artificial-intelligence_camera_model`.
- Modify contents: `mod/artificial-intelligence_q&a`, `q&a` is the brief English name of the file. For example, if you expect to modify the contents about `artificial intelligence camera model`, then the branch name should be: `mod/artificial-intelligence_camera_model`.

1.2.4 Q&A Guidelines

Please add new Q&A items and do updates according to the guidelines as follows:

General guideline:

- If you are going to add a new Q&A item, always remember to add a separate line after the previous one as "———————-".

For questions:

- Illustrate questions clearly, for example:
  - When flashing firmware to ESP32-S2, an error occurred as "A fatal error occurred: Invalid head of packet (0x50)" ? **(NOT Clear)**
  - When flashing firmware to ESP32-S2, an error occurred as "A fatal error occurred: Invalid head of packet (0x50)" . How to resolve such issue? **(Preferable)**
- Do not make the question too long. If this is the case, extract the main question as the title and describe the background and details below.

For answers:

- If there will be code in the text, use code box to separate it with the main text.
- If an answer only includes one sentence, there is no need to write a list, just use a regular paragraph.
- Use lists to separate items or to enumerate sequential items:
  - Use numbered lists for items that are in a required order (such as step-by-step instructions) or for items that are referred to by item number.
  - Use bulleted lists for items that are in no required order.
- Provide introductory phrase before a list to indicate the meaning or purpose of the list, and place a colon ":" at the end of it.
- If two items are alternatives, use a bullet list (not numbered list) and indicate their relationship in the introductory phrase.
- Always add two spaces before an listed item or paragraph in the answer.
- When a separate notice or explanatory paragraph follows a item, indent that separate material to the text of the parent list item.
- Follow list punctuation rules described in Espressif Manual of Style, Section Punctuation in Lists.

For more guidance on the rules of list formatting, please refer to ESP-Docs > Writing Documentation. Please refer to the text and image example templates below.

Q&A example

```
---------------

Does Espressif have any AI image recognition products?
```

(continues on next page)

(continued from previous page)

```
------------------------------------------------------

  Yes, we already have the ESP-EYE development board. With ESP32 as its␣
↪main control chip, ESP-EYE supports various types of cameras, such as␣
↪0v2640, 3660, 5640 and etc.
```

Q&A figure example

```
--------------

curses.h: No such file or directory?
-----------------------------------------

Screenshot: support ESP8266 chip, but ESP8266_RT

.. figure:: _static/application-solution/android-application/case_two_
↪kconfig_error.png
    :align: center
    :width: 900
    :height: 100

Solution: sudo apt-get install libncurses5-dev
```

1.2.5 Local build environment

- Use ubuntu or Debian system as test environment, and configure your python version to `3.7`.
- It is recommended to use python virtual environment or docker environment.

```
# Install python3.7 and virtual environment

sudo apt-get install python3.7 python3.7-venv

# Create virtual environment

python3.7 -m venv ~/.pyenv3_7

# Activate virtual environment

source ~/.pyenv3_7/bin/activate

# Upgrade pip

pip install --upgrade pip

# Install pip component

pip install -r docs/requirements.txt

# build the Chinese version

cd docs/cn/ && make html && cd -

# Build the English version

cd docs/en/ && make html && cd -

# Exit virtual environment
```

(continues on next page)

(continued from previous page)

```
deactivate
```

1.2.6 Commit message standards

Please add commit messages on your branch to explain what you have added/modified/deleted. Each commit has one message, for example:

```
artificial-intelligence: add esp-eye support those camera models

1. esp-eye support those camera models.
```

The first line of the commit message should be like "Q&A category: add/fix/modify/delete something". And this line should be started with the file name you updated, for example:

`artificial-intelligence: esp-eye support those camera models.`

If more information should be added into the commit message, please add it in the later commits after the first line.

A good commit message should tell why this update came up, thus making others get to know about this project when reading these commit logs. It may seem like a waste of time to write a good commit message, but it will be useful for you when trying to know why something changed.

1.2.7 Submit a merge request

Once your updates finished, you can conduct the first commit of your branch. Please add more commits if you need to do further updates. After finishing all the commits on this branch, you are ready to submit a merge request.

We use the github "Merge Requests" function to merge your branch into the master, the steps include:

1. Push your branch to the github repository;
2. Go to esp-faq and click "New pull request";
3. Select the branch that you created and waited for merge, and fill detailed information in the "Merge Request".

Please see IDF Contribution Guide.

Merge request specifications

- Title:

```
add: a brief overview
```

- Description:
  Describe the updates of this merge request in points.
- For example:

Title:

```
artificial-intelligence: add esp-eye support those camera models.
```

Description:

```
1. add esp-eye support those camera models.
```

Chapter 2

Development environment

2.1 IDE plugins

2.1.1 How to add ESP32 development board on Arduino IDE?

- For installation instructions of Arduino-ESP32, please refer to arduino-ide getting started.
- For instructions on how to add development boards on Arduino IDE, please refer to arduino Cores.

2.1.2 When using the Arduino IDE development platform, how to read the MAC address of the Wi-Fi that comes with ESP32?

- Please refer to the Arduino-ESP32 Development Framework.
- Use "WiFi.macAddress()" to obtain the MAC address of ESP32's Wi-Fi.
- Please also refer to the WiFiClientStaticIP example.

2.1.3 How to use the flash download tool to flash the bin file generated by Arduino to ESP32?

- Please navigate to `File` > `Preferences` > `Show verbose output during`, and check `compilation`. After compilation succeeded, a Python flashing command will be printed with the bin file to be flashed and the corresponding flashing address. For details, please refer to Arduino IDE Software Compilation Environment.
- Download the flash download tool, select the bin file when using the tool to flash, and enter the corresponding flashing address. For more information, please refer to Flash DownLoad Tool User Guide.

2.1.4 How to update the esp32-arduino library version?

- Under the path `Tool` > `Board` > `BoardManager`, search for `ESP32` to select and install the desired version of the esp32-arduino library. For more details, please refer to Installing ESP32 Arduino Software Development Environment on Windows.

2.1.5 Can ESP32-SOLO-1 be developed using the Arduino software?

- Currently, the Arduino software development environment only supports the operation of ESP32 dual-core chips and does not support the operation of ESP32 single-core chips. Therefore, ESP32-SOLO-1 is not yet supported for development on Arduino software.
- If you prefer to use Arduino to build your code, you can use the arduino-esp32 library as an ESP-IDF component for development and testing. For more details, please refer to How to use the arduino-esp32 library as an ESP-IDF component?.

2.2 Debugging

2.2.1 What is the serial port name of ESP devices ?

The serial port name is usually assigned by the operating system, and different operating systems and devices may have different serial port names. Common ones are as follows:

- In Windows system: COM*
- In Linux system: - UART: /dev/ttyUSB* - USB: /dev/ttyACM*
- In macOS system: /dev/cu.usbserial-*

2.2.2 How to block debugging messages sent through UART0 by default in ESP32?

The ESP32 log printing is divided into three levels: the first stage (ROM) bootloader chip startup log, the second stage bootloader log, and the application (app.bin) running log. By default, ESP32 outputs debug information through UART0, which can be turned off by following the steps below:

- Disable the first stage (ROM) bootloader log: Ground GPIO15, and the serial log output of the first stage (ROM) bootloader will be blocked upon power-up.
- Turn off the second stage bootloader log: In `menuconfig`, select `Bootloader config` > `Bootloader log verbosity` > `No output`.
- Turn off application logs: In `menuconfig`, select `Component config` > `Log output` > `Default log verbosity` > `No output`.
- Completely turn off UART0 as console output:
  - For IDF 4.3 and earlier, go to `menuconfig` > `Component Config` > `Common ESP-related` > `Channel for console output`, and select `None`.
  - For IDF 4.4 and later, go to: `menuconfig` > `Component config` > `ESP System Settings` > `Channel for console output`, and select `None`.

2.2.3 How to modify the default method of RF calibration in ESP32?

- During RF initialization, the partial calibration solution is used by default. Go to menuconfig and enable the `CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE` option.
- If the boot time is not critical, the full calibration solution can be used instead. Go to menuconfig and disable the `CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE` option.
- It is recommended to use the **partial calibration** solution, which ensures less boot time and enables you to add the function of erasing RF calibration information in NVS so as to trigger the full calibration operation.

For detailed information, please refer to the RF Calibration documentation.

2.2.4 How to modify the default method of RF calibration in ESP8266?

During RF initialization, the partial calibration solution is used by default, in which the value of byte 115 in esp_init_data_default.bin is `0x01`. The initialization only takes a short time. If the boot time is not critical, the full calibration solution can be used instead.

For NONOS SDK and earlier versions of RTOS SDK 3.0:

- Call system_phy_set_powerup_option(3) in function user_pre_init or user_rf_pre_init.
- In phy_init_data.bin, modify the value of byte 115 to `0x03`.

For RTOS SDK 3.0 and later versions:

- Go to menuconfig and disable CONFIG_ESP_PHY_CALIBRATION_AND_DATA_STORAGE.
- If CONFIG_ESP_PHY_INIT_DATA_IN_PARTITION is enabled in menuconfig, please modify the value of byte 115 in phy_init_data.bin to `0x03`. If CONFIG_ESP_PHY_INIT_DATA_IN_PARTITION is disabled, please modify the value of byte 115 in phy_init_data.h to `0x03`.

If you use the default partial calibration solution, and want to add the function of triggering the full calibration operation:

- For NONOS SDK and earlier versions of RTOS SDK 3.0, please erase the RF parameters to trigger the full calibration operation.
- For RTOS SDK 3.0 and later versions, please erase the NVS partition to trigger the full calibration operation.

2.2.5 How to troubleshoot in ESP32 Boot mode ?

- The ESP32-WROVER* uses 1.8 V flash and PSRAM, which is `0x33` by default in boot status and `0x23` in download mode.
- Other modules use 3.3 V flash and PSRAM, which are `0x13` by default in boot status and `0x03` in download mode.
- For detailed information, please refer to Section Strapping Pins in ESP32 Series Datasheet. Taking `0x13` as an example, the pins are as follows:

Pins	GPIO12	GPIO0	GPIO2	GPIO4	GPIO15	GPIO5
Level	0	1	0	0	1	1

You can also refer to the Boot Mode Selection documentation directly.

* indicates that the product is in EOL status.

2.2.6 When debugging with ESP32 JLINK, an ERROR occurs as: No Symbols For Freertos. How can I resolve such issue?

This issue will not affect actual operations. For solutions, please go to the ST Community.

2.2.7 How to monitor the free space of the task stack?

The function `vTaskList()` can be used to print the available space of the task stack regularly. For detailed information, please refer to CSDN Blog.

2.2.8 Is it possible to use JTAG to debug with ESP32-S2 ?

Yes. For detailed information, please refer to ESP32-S2 JTAG Debugging. For specific steps, refer to ESP32-S3 JTAG Debugging User Guide.

2.2.9 How to modify the log output without changing the output level of menuconfig ?

To modify the log output without changing the output level of menuconfig, you can use the `esp_log_level_set()` function. This function allows you to set the log level for a specific module or subsystem, rather than changing the global log level.

For example, to set the log level for the network module to `ESP_LOG_DEBUG`, you can use the following code:

```
esp_log_level_set("network", ESP_LOG_DEBUG);
```

For more information about this functionality, please refer to Logging library.

2.2.10 ESP8266 enters boot mode (2,7) and hits a watchdog reset. What could be wrong?

- Please make sure that when ESP8266 boots, the strapping pins are held in the required logic levels. If externally connected peripherals drive the strapping pins to an inappropriate logic level, ESP8266 may boot into a wrong mode of operation. With the absence of a valid program, the WDT may then reset the chip.
- Thus, in design practices, it is recommended to only use the strapping pins for input to high resistive external devices so that the strapping pin is not forced high/low at power-up. For more information, please refer to ESP8266 Boot Mode Selection.

2.2.11 When using the ESP-WROVER-KIT board with OpenOCD, an error occurred as `Can't find board/esp32-wrover-kit-3.3v.cfg`. How can I resolve such issue?

- With 20190313 and 20190708 versions of OpenOCD, please use instruction `openocd -f board/esp32-wrover.cfg`.
- With 20191114 and 20200420 (2020 later versions) versions of OpenOCD, please use instruction `openocd -f board/esp32-wrover-kit-3.3v.cfg`.

2.2.12 The RTC_watch_dog keeps resetting during ESP32 SPI boot. What could be the reason?

- Reason: The flash has a requirement for time interval between VDD_SDIO power-up and the first access. For example, GD' s 1.8 V flash requires 5 ms of time interval, while the time interval of ESP32 is about 1 ms (XTAL frequency is 40 MHz). Under such condition, the flash access will fail and either timer watchdog reset or RTC watchdog reset is triggered, depending on which one is triggered first. The threshold for RTC watchdog reset is 128 KB cycle, while the threshold for timer watchdog reset is 26 MB cycle. Taking the 40 MHz XTAL clock as an example, when the frequency of RTC slow clock is greater than 192 KHz, an RTC watchdog reset will be triggered first, otherwise a timer watchdog reset will be triggered. VDD_SDIO will be continuously powered when the timer watchdog is reset, so there will be no problem in accessing flash and the chip will work normally. When the RTC watchdog is reset, the VDD_SDIO power supply will be disabled and the access to flash will fail, resetting the RTC_watch_dog continuously.
- Solution: When an RTC watchdog reset occurs, the power supply to VDD_SDIO is disabled. You can add a capacitor to VDD_SDIO to ensure that the voltage of VDD_SDIO does not drop below the voltage that the flash can tolerate during this period.

2.2.13 How to obtain and parse coredump with ESP32?

- To obtain the 64 KB coredump file from the firmware, you need to know its offset from the partition table. Assuming the offset is `0x3F0000`, run the following command to read the firmware:

```
python esp-idf/components/esptool_py/esptool/esptool.py -p /dev/
↪ttyUSB* read_flash 0x3f0000 0x10000  coredump.bin
```

- Use the coredump reading script to convert the file obtained at the first step into readable messages. Assuming the coredump file is coredump.bin and the elf file is hello_wolrd.elf, run the following command to convert the file:

```
python esp-idf/components/espcoredump/espcoredump.py info_corefile -t␣
↪raw -c coredump.bin hello_world.elf
```

For more information, please refer to the Core Dump documentation.

2.2.14 How to do RF performance test with ESP32, ESP8266, and ESP32S2?

Download the ESP RF Test Tool and refer to the RF Test Guide for test instructions.

2.2.15 What are the reasons for not being able to recognize ESP devices under the Win10 system?

- Check if any security protection software has been activated.
- Check if the device is identified in the Linux virtual subsystem of Win10.
- If the device cannot be identified only in Win10 system, go to Device Manager to see whether such device exists (e.g., COM x). If the answer is still no, please check your cable and driver.
- If the device cannot be identified only in Linux virtual subsystem, taken VMWare as an example, please go to `Settings` > `USB Controller` and select `Show all USB input devices`.

2.2.16 One error occurred with ESP32 as `Core 1 panicked (Cache disabled but cache memory region accessed)`. What could be the reason?

Reasons:

- During the time when cache is disabled (e.g., when using the API spi_flash to read/write/erase/map the SPI flash), an interrupt is generated and the interrupt program accesses the flash resources.
- It is usually because the processor called programs from the flash and used its constants. One important thing is that since the Double variable is implemented through software, thus when this kind of variable is used in the interrupt programs, it is also implemented in the flash (e.g., forced type conversion operation).

Solution:

- Add an IRAM_ATTR modifier to the accessed function during interrupt
- Add an DRAM_ATTR modifier to the accessed constant during interrupt
- Do not use Double variable in the interrupt programs

For more information, please refer to the Fatal error documentation.

2.2.17 How to read the flash model information of the modules?

- For information about Espressif modules or chips. please refer to Flash Download Tool. For more details, please refer to the chipInfoDump Tab.
- In a Linux environment, you can read it through the python script esptool.

```
esptool.py -p /dev/ttyUSB* flash_id
```

2.2.18 When debugging the Ethernet Example in ESP-IDF, the following exception log appears. How can I resolve such issue?

```
emac: Timed out waiting for PHY register 0x2 to have value 0x0243(mask␣
↪0xffff). Current value:
```

You can refer to the following configurations of the development board. Please see the schematics for details:

- CONFIG_PHY_USE_POWER_PIN=y
- CONFIG_PHY_POWER_PIN=5

2.2.19 I found a "Brownout detector was triggered" failure on my ESP32. How to resolve such issue?

- ESP32 has a built-in brownout detector which can detect if the voltage is lower than a specific value. If it happens, the detector will reset the chip to prevent unintended behavior.
- This message may be reported in various scenarios, while the root cause should always be that the chip with a power supply has momentarily or permanently dropped below the brownout threshold. Please try replacing stable power supply and USB cable, or installing capacitor on power supply terminals of your module.
- For products powered by batteries, please check the power-on sequence, replace a battery with a higher current, or try to increase the capacitance of the power supply.
- Apart from the above solution, you can also try to configure the reset threshold value or disable the brownout detector. For more information, please refer to config-esp32-brownout-det.
- For ESP32 power-up and reset timing descriptions, see ESP32 Series Datasheet.

2.2.20 After ESP32 imported the protocol_examples_common.h header file, the file cannot be found while compiling. What could be the reason?

- Please add "set(EXTRA_COMPONENT_DIRS $ENV{IDF_PATH}/examples/common_components/protocol_examples_com in CMakeLists.txt under the project.
- For more information, please refer to the Build system documentation.

2.2.21 When using ESP8266 NonOS v3.0 SDK, the following error occurred. What could be the reason?

```
E:M 536    E:M 1528
```

Any error logs beginning with `E:M` indicates insufficient memory.

2.2.22 When using flash_download_tool to flash firmware to the ESP8266 module, how to solve the following error?

```
ESP8266 Chip efuse check error esp_check_mac_and_efuse
```

- Potential reasons:
  - `efuse check error` indicates that the internal eFuse parameter area of the chip has been unintentionally modified. Typically, eFuse stores critical information, such as chip configuration and MAC address. If the eFuse is damaged, it will disable the chip.
  - Generally, eFuse damage would be caused by overvoltage or static electricity.
- Suggestions:
  - Monitor voltage fluctuations during power-up and power-down processes.
  - The eFuse functionality has been enhanced in ESP32-C3/ESP32-C2 chips. You may consider replacing with the relevant products in the future.

2.2.23 When upgrading from ESP-IDF v4.4 to v5.0 and above, the error *esp_log.h:265:27: error: format '%d' expects argument of type 'int', but argument 6 has type 'uint32_t' {aka 'long unsigned int' } [-Werror=format=]265 | #define LOG_COLOR(COLOR) "033[0;" COLOR "m"* is reported. How can this be resolved?

- This error is caused by Espressif toolchain changes. For specific reasons and solutions, please refer to the Migration Guide: From 4.4 to 5.0.
- If you decide to ignore this error (not recommended), you can add `target_compile_options(${COMPONENT_LIB} PRIVATE -Wno-pointer-sign -Wno-format)` in the corresponding cmake of the file where the compilation error occurred.

2.2.24 Do ESP32 series products support the use of JTAG functionality in a boundary scan environment? Where can I download the BSDL files?

Due to hardware limitations, currently ESP32 series products do not support boundary scan functions, therefore JTAG can not be used in a boundary scan environment, and there are no BSDL files.

2.3 Environment Setup

2.3.1 When setting up ESP32-S2 environment using command `idf.py set-target esp32s2`, an error occurred as `Error: No such command 'set-target'`. What could be the reason?

- The ESP-IDF is adapted to ESP32-S2 from release/v4.2, thus setting up ESP32-S2 environment in previous versions will cause errors. In this case, when using command `idf.py set-target esp32s2`, there will be error as `Error: No such command 'set-target'`. It is recommended to perform tests and development on ESP32-S2 using ESP-IDF release/v4.2 and later versions. For more information, please refer to ESP32-S2 Get Started.
- For information on the support of different ESP chips by different ESP-IDF versions, please refer to ESP-IDF Release and SoC Compatibility.

2.3.2 When installing ESP-IDF version master using ESP-IDF Tools 2.3 in Windows system, an error occurred as `Installation has failed with exit code 2`. What could be the reason?

This is related to the bad network environment. The Github repository cannot be downloaded smoothly under such network environment, causing SDK download failure on your PC. If you encounter Github access problems, it is recommended to use the **offline** version of the latest ESP-IDF Windows Installer.

2.3.3 On Windows, when setting up the environment using esp-idf-tools, the following error occurs when running `make menuconfig`:

```
-- Warning: Did not find file Compiler/-ASM Configure
-- Configuring incomplete, errors occurred!
```

This is because the system could not find the project to be compiled. You need to change directory to the ESP-IDF project before running commands to configure and compile the project. For example, to build the project hello world, go to `esp-idf/examples/get-started/hello_world` before running the commands.

2.3.4 During the installation process of esp-idf-tools on Windows, an exception occurs with the Python tool:

```
Installation has failed with exit code 1
```

- This error is usually caused by an unstable network environment. It is recommended to check the "Download via gitee" (Mirror Settings) option when installing tools to avoid issues with poor GitHub network connections.
- For a more reliable installation option, consider using the offline version of the Windows installation tool downloaded from Espressif download server, which includes all required software components and does not rely on real-time downloads during the installation process.

2.3.5 What should I do if I get `Download failed: security channel support error` when installing build environment in the Windows system?

This is because the Windows system has disabled the default support for SSl3.0.

Solution: Go to *Control Panel* and find `Internet option`, select `Advanced`, and check the `use SSL 3.0` option.

2.3.6 When executing export.bat in Windows system, what should I do if I get CMake and gdbgui version errors?

```
C:\Users\xxxx\.espressif\tools\cmake\3.16.4\bin
The following Python requirements are not satisfied:
gdbgui>=0.13.2.0
```

This is because the upstream gdbgui has been updated, thus it is not compatible with the low version of python. The current solution is to manually modify the root file `requirements.txt` in ESP-IDF by changing the description of gdbgui version to `gdbgui==0.13.2.0`.

2.3.7 Errors occurred when using idf.menuconfig and idf.build after updating the ESP-IDF version from v3.3 to the latest one:

- Please refer to the Quick Start Guide to set up the environment again.
- Remove build directory `build` and configuration file `sdkconfig` under the hello_world directory.

2.3.8 How to configure `PATH` and `IDF_PATH` when developing ESP32 and ESP8266 simultaneously?

- For `PATH`, there is no need to do extra configurations. You can put them together as: export PATH=" $HOME/esp/xtensa-esp32-elf/bin:$HOME/esp/xtensa-lx106-elf/bin:$PATH" .
- For `IDF_PATH`, you can specify it for separate chips as:
  In ESP32 related projects, use `IDF_PATH = $(HOME)/esp/esp-idf`. In ESP8266 related projects, use `IDF_PATH = $(HOME)/esp/ESP8266_RTOS_SDK`.

2.3.9 Do I need to use command `idf.py set-target` every time when switching to another project?

When building the project with `idf.py build`, the target is determined as follows:

1. If the build directory `build` already exists, the system will use the target the project was previously built for. It is stored in CMakeCache.txt file in the `build` directory.
2. Alternatively, if the build directory doesn't exist, the system will check if the `sdkconfig` file exists, and use the target specified there.
3. If both the build directory and `sdkconfig` file exist with different targets specified, the system will report an error. This shouldn't happen normally, unless `sdkconfig` was changed manually without deleting the build directory.

4. If neither `sdkconfig` file nor build directory exists, it can be considered to use `IDF_TARGET` to set the target as a CMake variable or environment variable. If this variable is set and is different from the target specified in `sdkconfig` or in the build directory, the system will also report an error.
5. Finally, if `sdkconfig` doesn't exist, build directory doesn't exist, and the target is not set via `IDF_TARGET`, then the system will use the default value. The default value can be set in `sdkconfig.defaults`.
6. If the target isn't set using any of the above methods, then the system will build for ESP32 target.

To answer your question:

- `idf.py set-target` stores the selected target in the project's build directory and `sdkconfig` file, not in the terminal environment. So, once the project is configured and built once for a certain target, if you switch to a different directory and build another project, then come back, the target will not change, and will be the same as previously set for this project. And it's not necessary to run `idf.py set-target` again other than to switch to a different target.
- If you want to make the project built for certain target by default, add `CONFIG_IDF_TARGET="esp32s2"` to the `sdkconfig.defaults` file of the project. After this, if `sdkconfig` file doesn't exist and build directory doesn't exist, idf.py build command will build for that target specified in `sdkconfig.defaults`.
- `idf.py set-target` command can still be used to override the default target set in `sdkconfig.defaults`.

2.3.10 How to know the version of ESP-IDF, is it recorded in a certain document?

- Command line: You can obtain the version number by inputting `idf.py --version` in the terminal with an IDF environment.
- CMake script: You can obtain the version number through the variable `${IDF_VERSION_MAJOR}.${IDF_VERSION_MINOR}.${IDF_VERSION_PATCH}`.
- Code compilation: You can obtain the version number by calling `esp_get_idf_version` during code compilation or directly using the macro definition of version in `components/esp_common/include/esp_idf_version.h`.

2.3.11 How to optimize ESP-IDF compilation in Windows environment?

Please add the ESP-IDF source directory and the compiler directory `.espressif` to the antivirus software's exclusion list.

2.3.12 Is there an esptool that can be used directly on Windows?

You can go to esptool > Releases, and download the Windows version of the esptool tool from the Asset column on the dropdown page.

2.3.13 What could be the cause of the error `KeyError: 'idfSelectedId'` when running `. /install.sh`?

- This is because ESP-IDF v5.0 or above is installed on your system. You can check the configuration in the `~/.espressif/idf-env.json` file.
- This error can be solved by running `rm -rf ~/.espressif/idf-env.json`.

2.3.14 When running `demo`, the package manager component dependency cannot be pulled, with the failure `Invalid manifest format`, `Invalid dependency format`, and `unknown keys in dependency details: override_path`. What could be the reason?

This is due to missing component dependencies, and it requires an update to the *component-manager*. The command instruction is `pip install --upgrade idf-component-manager`.

2.3.15 After installing the ESP-IDF CMD environment using the ESP-IDF v4.4.8-Offline Installer package, why does the following compilation error occur when directly compiling the hello_world example?

```
[1050/1065] Building C object esp-idf/main/CMakeFiles/__idf_main.dir/main.
↪c.obj
FAILED: esp-idf/main/CMakeFiles/__idf_main.dir/main.c.obj
D:\esp\Espressif\tools\xtensa-esp32-elf\esp-2021r2-patch5-8.4.0\xtensa-
↪esp32-elf\bin\xtensa-esp32-elf-gcc.exe: error: @-file refers to a␣
↪directory
[1058/1065] Building C object esp-idf/wifi_provisioning/CMakeFiles/__idf_
↪wifi_provisioning.dir/src/scheme_softap.c.obj
ninja: build stopped: subcommand failed.
ninja failed with exit code 1
```

- According to the logs, an error occurred while caching the `build/esp-idf/main/CMakeFiles/__idf_main.dir/ main.c.o.bj` file during the compilation process. This file is generated when ccache calls the compiler and is related to the compilation cache. This issue has been resolved in versions 5.0 and later.
- In the ESP-IDF CMD environment of version v4.4, please use the `idf.py --no-ccache build` command to build the projects.

2.3.16 How to fix the following error when building a project and pulling components?

```
CMake Error at C:/Espressif/frameworks/esp-idf-v5.1.2/tools/cmake/build.
↪cmake:540 (message):
  ERROR: Cannot establish a connection to the component registry. Are you
  connected to the internet?

  URL:
  https://components-file.espressif.com/components/espressif/esp_lcd_
↪gc9a01.json
```

- Create a file named `idf_component_manager.yml` in the `IDF_TOOLS_PATH` directory with the following content:

```
profiles:
  default:
    storage_url:
      - "https://components-file.espressif.cn"
```

- For more details, refer to idf_component_manager.yml Configuration File.

2.3.17 How to mirror esp-idf SDK and related toolchains from GitHub to domestic sources on Windows?

- In the **ESP-IDF CMD terminal** environment, you can run the following commands:
  - **ESP-IDF repository (Git) mirror**

```
git config --global url."https://jihulab.com/esp-mirror/".insteadOf
↪"https://github.com/"
```

To remove the mirror settings:

```
git config --global --unset url."https://jihulab.com/esp-mirror/".insteadOf
git config --global --list
```

  - **ESP-IDF tools mirror**

```
set IDF_GITHUB_ASSETS=dl.espressif.com/github_assets
```

To remove the mirror settings:

```
set IDF_GITHUB_ASSETS=
echo %IDF_GITHUB_ASSETS%
```

  - **PyPI mirror**

```
python -m pip config set global.index-url https://mirrors.aliyun.com/pypi/
↪simple
# or:
# python -m pip config set global.index-url https://pypi.tuna.tsinghua.edu.
↪cn/simple  # Tsinghua mirror
# or:
# python -m pip config set global.index-url https://pypi.mirrors.ustc.edu.
↪cn/simple  # USTC mirror
```

To remove the PyPI mirror settings:

```
python -m pip config unset global.index-url
python -m pip config list
```

- In the **PowerShell terminal** environment, you can run the following commands:
  - The Git repository mirror and PyPI configuration commands are the same as in **CMD** above;
  - **ESP-IDF tools mirror**

```
$env:IDF_GITHUB_ASSETS="dl.espressif.com/github_assets"
```

To remove the mirror settings:

```
Remove-Item Env:IDF_GITHUB_ASSETS
$Env:IDF_GITHUB_ASSETS
```

2.3.18 Which operating systems and development environments/IDEs are supported by Espressif chips?

- Official framework and environment:
  - ESP-IDF, supports Windows, Linux, macOS. It can be used with the command line, or in conjunction with the Visual Studio Code Plugin or Eclipse / Espressif-IDE plugins for graphical development;
  - Arduino IDE, based on the Arduino core that encapsulates some of the ESP-IDF capabilities, is easier to get started with.
- Other common solutions: CLion, CircuitPython, MicroPython, PlatformIO, Toit, UIFlow, Wokwi, etc. For details, see ESP-Techpedia.

2.4 Firmware Update

2.4.1 How does the host MCU flash ESP32 via serial interfaces?

- For the related protocol, please refer to ESP32 Serial Protocol. For the corresponding documentation, please refer to Serial Protocol.
- For code examples, please refer to esp-serial-flasher.

2.4.2 How to download firmware for ESP32 series modules using the USB-Serial tool?

Refer to Hardware Requirements for Entering Download Mode on Different ESP Chips.

2.4.3 How to flash firmware on macOS and Linux?

- For the Apple system (macOS), you can use esptool downloaded via brew or git to flash firmware.
- For the Linux system (e.g., Ubuntu), you can use esptool downloaded via apt-get or git to flash firmware.

2.4.4 Does ESP32 support programming using JTAG pins directly?

ESP32 supports flashing programs directly using JTAG pins. Please refer to Upload the application to be debugged. For detailed steps, see Using the ESP32 JTAG interface for firmware download in a Windows environment.

Note: For mass production flashing, it is recommended to use the flash download tool to flash the firmware. For detailed instructions, see Choosing the appropriate burning platform.

2.4.5 Does ESP_Flash_Downloader_Tool support customized programming control?

- The GUI tool is not open-sourced and does not support embedded executive script.
- The low-level component esptool is open-sourced and can be used to perform all functions such as flashing and encryption. It is recommended to conduct secondary development based on this component.

2.4.6 Can I enable the Secure Boot function for ESP32 via OTA?

- It is not recommended to enable Secure Boot through OTA, as it poses operational risks and requires multiple OTA firmware updates.
- Since the Secure Boot function is in Bootloader, please update Bootloader first to enable this function.
  1. First, check whether the partition table of your current device can store the Bootloader with Secure Boot enabled.
  2. Then, update an intermediate firmware which can be written in the Bootloader partition. By default, the Bootloader partition cannot be erased or written, you need to enable them via *make menuconfig*.
  3. Sign the intermediate firmware and upgrade it to the target device through OTA. Then upgrade the Bootloader of this firmware and the signed new firmware through OTA.
  4. If there are situations such as powered-down or network break-down and restart during the Bootloader OTA process, the device will not be booted and needs to be re-flashed.

2.4.7 How to resolve the following error that occurred when flashing firmware to ESP32-S2 based on ESP-IDF v4.1?

```
esptool.py v2.9-dev
Serial port /dev/ttyUSB0
Connecting....
Chip is ESP32S2 Beta
Features: Engineering Sample
Crystal is 40MHz
MAC: 7c:df:a1:01:b7:64
Uploading stub...
Running stub...

A fatal error occurred: Invalid head of packet (0x50)
esptool.py failed with exit code 2
```

Solution:

If you are using ESP32-S2 instead of ESP32-S2 Beta, please update ESP-IDF to v4.2 or later versions.

Notes:

- ESP-IDF v4.1 only supports ESP32-S2 Beta, which is not compatible with ESP32-S2.
- The version of esptool that comes with ESP-IDF v4.1 is v2.9-dev, which only supports ESP32-S2 Beta as well.
- Both ESP-IDF v4.2 and its esptool v3.0-dev support ESP32-S2 series chips.

2.4.8 How to download firmware based on ESP-IDF using flash_download_tool?

- Please refer to get-started-guide when building an ESP-IDF project for the first time.
- Taken hello-world example for instance, run `idf.py build` (supports ESP-IDF v4.0 and later versions, for versions before v4.0, please use `make`). After building, the following flash command for the bin file will be generated:

```
#Project build complete. To flash, run this command:
../../../components/esptool_py/esptool/esptool.py -p (PORT) -b 921600␣
↪write_flash --flash_mode dio --flash_size detect --flash_freq 40m␣
↪0x10000 build/hello-world.bin  build 0x1000 build/bootloader/bootloader.
↪bin 0x8000 build/partition_table/partition-table.bin
or run 'idf.py -p PORT flash'
```

You can use flash_download_tool to flash according to the bin file and flash address prompted by this command.

2.4.9 What is the communication protocol for flashing ESP chips?

- ESP flashing protocol specification: Serial Protocol.
- Serial protocol Python implementation: esptool.
- Serial protocol C language implementation: esp-serial-flasher.

2.4.10 How to program ESP32-C3's firmware offline?

- Currently, no tool supports the offline programming of ESP32-C3's firmware. However, the official Flash Download Tools can directly download binary firmware and support mass production download mode for up to eight ESP32-C3 devices at the same time.
- In addition, we also provide Test Fixture for mass production, which supports up to four ESP32-C3 modules to download firmware simultaneously.

2.4.11 How to set the flash SPI mode to QIO mode on ESP32?

Go to menuconfig and set it through `Serial flasher config` > `Flash SPI mode`, the corresponding API is esp_image_spi_mode_t.

2.4.12 After downloading the program and powering on EPS8266, the serial port printed the following log. What is the reason?

```
ets Jan  8 2013,rst cause:1, boot mode:(7,7)
waiting for host
```

waiting for host means the Boot is in SDIO mode, indicating that GPIO15 (MTDO) is pulled up (HIGH). Please refer to ESP8266 Boot Mode Description.

2.4.13 What are the Espressif module programming tools?

- For Espressif programming software, please go to Flash Download Tools. Installation-free GUI tools are for the `Windows` environment only.
- Espressif programming tool esptool is written based on *Python* with open-source code, supporting secondary development.

2.4.14 What is the difference between the Factory mode and Developer mode of the flash download tool?

- Factory mode supports multi-channel downloads, while Developer mode only supports single channel.
- The path of bin files under the Factory mode is relative, while under Developer is absolute.

2.4.15 The ESP32-C3 chip should be able to conduct firmware download through USB, but I failed to do so under ESP-IDF v4.3. So, how can I use USB for firmware download?

You need to compile under ESP-IDF v4.4 or above versions. After pulling the latest branch and updating the IDF tool, you can compile normally and use USB for download. Please refer to usb-serial-jtag-console.

2.4.16 Why does the programming fail for the jig with a 4-port hub in factory mode?

:CHIP: ESP32 | ESP8266 :

- It is because Espressif products complete the calibration operation by transmitting some packets when starting up. This operation requires a 3.3 V voltage and a guaranteed peak current of 500 mA. Therefore, when it comes to more than one port, there will be situations where the computer cannot program or the programming is interrupted due to the insufficient power supply of the computer' s USB when programming via connecting to a computer' s USB. It is recommended to use the hub for programming and supply power to the hub in the meantime.

2.4.17 I' m using an ESP32-WROVER-B module to download the AT firmware via the flash download tool. However, an error occurred after writing to flash. But the same operation succeeded when replacing the module with ESP32-WEOVER-E, what is the reason?

- The ESP32-WROVER-B module leads out the SPI flash pin, but the ESP32-WROVER-E module does not. Please check whether the SPI flash pin of the ESP32-WROVER-B module is reused by other external application circuits.
- Connecting the CMD pin of the SPI flash in ESP32-WROVER-B to GND will cause the flash to fail to start. The following error log will be printed:

```
rst:0x10 (RTCWDT_RTC_RESET),boot:0x1b (SPI_FAST_FLASH_BOOT)
flash read err, 1000
ets_main.c 371
ets Jun 8 2016 00:22:57
```

2.4.18 Why can' t the Flash Download Tools be used to reflash the firmware on a device that has enabled flash encryption but not disabled the download mode?

CHIP: ESP32 | ESP32-S2

- The default configuration of the flash download tool has enabled eFuse verification. If you want to reflash the firmware of a device that has already enabled flash encryption, please modify the following configuration:
  - Modify the default configuration in the `esp32 > security.conf` file, change `flash_force_write_enable = False` to `flash_force_write_enable = True`.
  - Modify the default configuration in the `esp32 > spi_download.conf` file, change `no_stub = False` to `no_stub = True`.
- Note: When reflashing the firmware on devices with flash encryption enabled, the reflashed firmware must use the same flash encryption key. If the keys do not match, the new firmware will not function properly.

2.4.19 When updating ESP32 firmware through the UART interface based on esptool serial port protocol, can I add a new app partition?

The actual partitioning of the flash primarily depends on the data in `partition_table.bin`. If `partition_table.bin` can be updated, the storage space for other data such as `bootloader.bin`, `app.bin`, etc., can be re-partitioned, thereby adding a new app partition.

2.4.20 I am using ESP8266 to download the firmware via the flash download tool. After downloading the firmware, there is no programming output log, and the serial port prints the following messages. What could be the reason?

```
ets Jan  8
2013,rst cause:1, boot mode:(3,7)
ets_main.c
```

- Please check whether the hardware wiring is correct. See Boot mode wiring instructions.
- Please check whether the download offset address of `bootloader.bin` is correct. The offset address downloaded from `bootloader.bin` of ESP8266 is "0x0". If the offset address is wrong, the flash cannot be started.

2.4.21 Why does my USB driver fail to be recognized by the Windows 7 system?

Please download and install the USB Serial JTAG driver manually for the Windows 7 system.

2.4.22 After using the ESP32-WROVER-E module to download the program, the following log is printed after powered on. What could be the reason?

```
rst: 0x10  (RTCWDT_RTC_RESET), boot:0x37 (SPI_FLASH_BOOT)
【2020-12-11 15:51:42 049】invalrd header: 0xffffffff
invalrd header: 0xffffffff
invalrd header: 0xffffffff
```

- Generally, it is because the GPIO12 was pulled high. It is recommended to pull it low and see the results. Please see ESP32 Boot Log Guide.

2.4.23 When using the Flash Download Tools to flash ESP32-C3 via USB, 8-download data fail occurs repeatedly. How can I solve it?

- Please erase the chip completely first before flashing.
- This problem has been solved in V3.9.4 and above versions.

2.4.24 On ESP32, the bootloader.bin of ESP-IDF v3.0 can not start app.bin of ESP-IDF v5.0. Why?

When booting the `app.bin` of ESP-IDF v5.0 with the `bootloader.bin` of ESP-IDF v3.0, it is necessary to enable the configuration option `idf.py menuconfig > Build type > [*] App compatible with bootloader and partition table before ESP-IDF v3.1` on ESP-IDF v5.0.

2.4.25 Does ESP32-C3 support disabling ROM code logs via OTA?

Yes. You can disable the ROM code log by enabling the *Boot ROM Behavior → Permanently change Boot ROM output → (X) Permanently disable logging* configuration in the software, and then update the firmware via OTA.

2.4.26 Will the operation of other tasks be affected when the chip is undergoing an OTA firmware upgrade (esp_ota_write())?

During the OTA process, the cache will be turned off when writing to the flash, which will affect peripheral interrupts and some SPI tasks. Therefore, it is not recommended to perform other tasks during this period.

2.4.27 How many devices can be flashed at once in factory mode using Flash Download Tool?

Currently, up to 20 devices are supported. You need to update to the latest version of the Flash Download Tool (3.7.8) and modify the configuration file `flash_download_tool\configure\esp32s3\multi_download.conf` as follows:

```
[MULTI_UI_CONFIG]
multi_col = 4
```

2.4.28 Does ESP32 OTA support resumable uploads?

Supported. Please refer to OTA Resumption.

2.4.29 When using esptool for host computer development, after increasing the download baud rate to above 1,152,000, there is no significant improvement in download speed. Is this a limitation imposed by esptool?

esptool itself does not limit the baud rate. As the baud rate continues to increase, the data processing time on both the host side and the chip side becomes the main bottleneck, making further baud rate increases ineffective at improving download speed. Currently, there are no more methods to further increase the download speed.

Chapter 3

application solution

3.1 Artificial intelligence

3.1.1 What types of cameras are supported on AI image recognition products?

With ESP32 as its main control chip, ESP-EYE supports various types of cameras, such as 0V2640, OV3660, OV5640, OV7725, etc. Please see esp32-camera Github.

For the ESP32-P4 series, please refer to esp-video-components to get the supported camera models.

3.1.2 Which versions of ESP-IDF are supported by ESP-WHO?

The subsequent supported versions will be updated on ESP-WHO Github.

3.1.3 Is there any information about the WeChat mini program of ESP-EYE?

For open source resources of ESP-EYE mini program demo, please check EspEyeForWeChat.

3.1.4 What languages are supported by the esp-skainet demo?

Only Chinese and English currently.

3.1.5 What model frameworks does ESP-DL support?

The quantization tool ESP-PPQ used by ESP-DL can directly quantize ONNX models. Pytorch and TensorFlow need to be converted to ONNX models first. Please make sure your model can be converted to an ONNX model.

3.1.6 Does ESP-DL support all models of the three platforms mentioned above (MXNet, Pytorch, and TensorFlow)?

All operators in the above models must be supported by ESP-DL. For supported operators, please refer to Operator Support State.

3.1.7 Can the model files of ESP-SKAINET be stored in the SD card?

Yes.

3.1.8 How to customize command words in ESP-SKAINET?

For custom command words, please refer to the Custom Command Word Method.

3.1.9 How to reduce the system footprint of AI speech models?

You can choose to turn off the three functions, namely AEC, AE, and VAD.

3.1.10 What is the difference between a 16-bit quantization model and an 8-bit quantization model?

The 16-bit quantization model has higher precision and more accurate results, while the 8-bit quantization model is more lightweight.

3.1.11 How does the AI voice model modify the number of microphone channels?

The number of microphone channels and the number of playback channels can be configured in the AFE.

3.1.12 How do I get the actual audio captured in the development board?

To obtain the actual audio, an SD card interface is required to store the audio files to the SD card.

3.1.13 Do you have relevant study materials for ESP-SR GitHub?

Please refer to ESP-SR User Guide.

3.1.14 Do you have relevant study materials for ESP-DL?

Please refer to the following tutorials:

- Hand Gesture Recognition with ESP-Deep Learning.
- Deploy Touch Handwritten Digit Recognition with ESP-DL.
- Pedestrian Detection.
- Face Detection.
- Face Recognition.
- Yolo11N Object Detection.
- Cat Face Recognition.
- Color Recognition.
- Lightweight Object Detection Framework ESP-Detection.

3.1.15 How does ESP32-S3 customize English command words for recognition?

- For MultiNet6, you need to prepare `commands_en.txt` to customize English command words. For MultiNet5, you need to use the `multinet_g2p.py` script to convert English command words into phonemes that can be recognized by multinet. For details, please refer to esp-sr/tool.

3.1.16 How to improve the detection efficiency of Yolo11?

- You can try quantizing the model with small resolution, but this may cause some loss in accuracy. For details, please refer to quantize_yolo11n.
- Use the lightweight detection model framework esp-detection.

3.2 AT

For ESP-AT FAQ, please go to ESP-AT User Guide.

3.3 Audio Application Framework

3.3.1 When using the VoIP feature of ESP-ADF, how to eliminate echo during calls between a mobile phone and an ESP32 device?

- Espressif provides an Acoustic Echo Cancellation (AEC) algorithm based on ESP32 and ESP32-S3 chips. For details, please refer to the algorithm examples.
- Note that the effect of AEC not only depends on software parameter configuration and debugging, but also on hardware design, such as distortion-free playback, noise-free recording, and problem-free echo reference signal, etc. For this part, it is recommended to refer to the Espressif ' s design on the ESP32-Lyrat-Mini development board and ESP32-S3-Korvo-2 development board.

3.3.2 Is there a reference example for accessing the big language model provided by Baidu?

Please refer to the dueros example.

3.3.3 Does the official VoIP example provided by Espressif support RTP?

- In ESP-ADF, the default VoIP provided is based on SIP, and the protocol utilizes RTP.
- You can use the VoIP example under Espressif SDK ESP-ADF.

3.3.4 Is the SIP protocol in ESP-ADF open-source?

The protocol is currently not open-source and is provided for external calls in the form of a library.

3.3.5 Can ESP-ADF examples implement the volume adjustment function of Bluetooth headsets?

Refer to example: pipeline_bt_sink

3.3.6 I want to use a sensor on the I2C of ESP32-LyraT. Is there an example on how to read I2C device data?

Please refer to the i2c example.

3.3.7 How to output 32-bit I2S audio data?

Please refer to the following code:

```
i2s_stream_cfg_t i2s_writer_cfg = I2S_STREAM_CFG_DEFAULT();
i2s_writer_cfg.type = AUDIO_STREAM_WRITER;
i2s_writer_cfg.stack_in_ext = true;
i2s_writer_cfg.task_core = 1;
i2s_writer_cfg.need_expand = true;
i2s_writer_cfg.expand_src_bits = 16;
i2s_writer = i2s_stream_init(&i2s_writer_cfg);
```

3.3.8 Why do I always get an error when compiling example/get-started/play-mp3 with ESP-ADF and ESP-IDF v4.1?

Error log: `fatal error: audio_type_def.h: No such file or directory`

- The file audio_type_def.h is located in the esp-adf-libs of ESP-ADF. If this file cannot be found during the compilation process, it indicates that ESP-ADF v2.4 may not have been correctly detected, especially the submodules may not have been updated.
- To correctly detect ESP-ADF v2.4, please follow the steps described in Update to a stable release version.
- Try executing the following commands and recompile.

```
cd $ADF_PATH
git fetch
git checkout v2.4
git submodule update --init --recursive
```

3.3.9 Where can I check the ESP-IDF version supported by ESP-ADF?

Please refer to the ESP-IDF version supported under ESP-ADF.

3.3.10 Will integrating DuerOS occupy the recording function of the ESP32-LyraT development board throughout the process?

The current design occupies the recording data throughout the process. However, you can enable the `multi_output` function of `I2S_stream` to output the recording data to the desired location through this channel.

3.3.11 Does ESP-ADF support customizing voice recognition wake-up words?

The voice training interface has not yet been opened. You can directly use the free wake-up word "Hi, Espressif". If you have customization requirements, please email sales@espressif.com for details.

3.3.12 Does ESP-ADF support running Alexa examples on the ESP32-LyraTD-MSC v2.1 development board?

- ESP-ADF does not directly support Alexa examples. For Alexa examples, please refer to esp-va-sdk.
- Please note that since the second half of 2024, Alexa has disabled the host-side (built-in) voice interface. If you need to connect to the Alexa voice ecosystem in the future, you can try using the ACK or Matter solution.

3.3.13 Regarding voice recognition on ESP32, can you recommend a suitable development board for localization?

We recommend using the ESP32-Lyrat-Mini development board or the ESP32-S3-Korvo-2 development board for localization.

3.3.14 Does ESP32 have a development board that supports both MIC and AUX pickup?

The ESP32-lyraT-4.3 development board supports both MIC and AUX pickup.

3.3.15 How to implement call function using the ESP32-LyraT development board?

You can refer to the VoIP example VoIP.

3.3.16 What is the maximum power of supported speakers for ESP32 series audio development board?

- ESP32 development board uses NS4150 PA by default, and its maximum power is 3 W.
- You may change the PA design according to needs.

3.3.17 Does Espressif's voice wake-up solution have specific requirements regarding environmental noise?

The current Espressif voice solution can meet the environmental requirements of a signal-to-noise ratio of less than 5 dB. For some fixed noise scenarios, it can even be less than 0 dB (need to be optimized for the actual product).

3.3.18 If there is an AUX input on the ESP32's AI development board, can the MIC still be used to pick up sound?

- The ESP-ADF development framework can choose various ways to pick up sound, including MIC input and Line-in.
- The pickup method selection is as follows:

```
typedef enum {
  AUDIO_HAL_CODEC_MODE_ENCODE = 1, /*! <select adc */      // MIC pickup
  AUDIO_HAL_CODEC_MODE_DECODE, /*! <select dac*/
  AUDIO_HAL_CODEC_MODE_BOTH, /*! <select both adc and dac */   //  MIC +␣
↪speaker
  AUDIO_HAL_CODEC_MODE_LINE_IN, /*! <set adc channel */,             //␣
↪microphone pickup
} Audio_hal_codec_mode_t;
```

- The pickup method configuration is as follows:

```
audio_board_handle_t board_handle = audio_board_init();
audio_hal_ctrl_codec(board_handle->audio_hal, AUDIO_HAL_CODEC_MODE_DECODE,␣
↪AUDIO_HAL_CTRL_START);   // If you want to pick up sound from the MIC,␣
↪modify this configuration option.
```

3.3.19 When designing an audio development board with the ESP32-WROVER-B module and ES8311, which pins can be selected for the MCLK clock?

- On the hardware side, MCLK can only use GPIO0, GPIO1, GPIO3 pins. You can check the IO_MUX table in the ESP32 Datasheet, where GPIO0 is used by default.
- You can refer to the design of the schematic of ESP32-LyraT-Mini.
- For allocation of pins, please refer to ESP32-LyraT-Mini V1.2 Hardware Reference.

3.3.20 Can the ESP32-WROVER-E module use a single I2S to realize simultaneous playback and recording?

Yes. You can refer to the ESP32-LyraT getting started guide.

3.3.21 Do Espressif modules support Spotify Connect?

:CHIP: ESP32 | ESP32-S2 | ESP32-S3 :

Not supported yet. It is recommended to consider using dlna, which can achieve a similar effect.

3.3.22 When running the korvo_du1906 example on an ESP32-Korvo-DU1906 board, a reboot caused the following error message: Guru Meditation Error: Core 0 panic'ed (IllegalInstruction). Exception was unhandled. How to resolve such issue?

- Please check the power supply.
- Provide power to the entire system. It is recommended to use at least a 5 V/2 A power adapter to ensure stable power supply.

3.3.23 Can ESP-DSP fft run 4096, 8192, and more samples?

Yes, it supports up to 32 K samples. The maximum value can be configured in menuconfig. For example, in the fft demo, the configuration steps are `idf.py menuconfig` > `Component config` > `DSP Library` > `Maximum FFT length` > `(*) 32768`.

3.3.24 How to connect a microphone to ESP32?

- You can connect I2S peripheral if it is a digital microphone.
- You can connect ADC peripheral if it is an analog microphone.
- Considering the recording quality, it is more recommended to use an external AD.

3.3.25 Does ESP32 support analog audio or digital audio output?

- ESP32 supports DAC analog audio output, which can be used to play simple audio such as prompt tones.
- ESP32 supports PWM analog audio output, which is slightly better than DAC. Demonstration code: esp-iot-solution.
- ESP32 also supports I2S digital audio output. For I2S configurable pins, please refer to ESP32 Datasheet > Chapter Peripherals and Sensors.

3.3.26 What audio formats does the ESP32 chip support?

ESP32 supports audio formats such as MP3, AAC, FLAC, WAV, OGG, OPUS, AMR, G.711, etc. For more information, please refer to the instructions under ESP-ADF SDK.

3.3.27 How to use the ESP32 chip to decode compressed audio?

For applications using the ESP32 chip to decode compressed audio, refer to the examples in the esp-adf/examples/recorder folder.

3.3.28 Where is the code example for ESP-LED-Strip?

The corresponding code examples are stored in ESP-ADF. Please refer to the led_pixels example.

3.3.29 Does ESP32 support online voice recognition?

Yes, it does. Please refer to the esp-adf/examples/dueros example.

3.3.30 Does ESP32 support volume adjustment of Bluetooth headphones?

Yes. ESP32 uses the Bluetooth AVRCP tuning protocol. You can test the function with the pipeline_a2dp_sink_stream example.

3.3.31 Can Wi-Fi and FFT be used simultaneously?

Wi-Fi and FFT can be used simultaneously. For example, Wi-Fi functionality can be directly added to the rhythm light example with FFT functionality.

3.3.32 What does the RECORD_HARDWARE_AEC macro mean when it is True or False in the ESP-ADF SDK?

The value of RECORD_HARDWARE_AEC macro being True or False indicates whether the development board has a hardware AEC (Acoustic Echo Cancellation) circuit. If it is True, the development board has a hardware AEC circuit. If it is False, the development board does not have a hardware AEC circuit.

3.3.33 Are there any recommended product models and application references for developing an outdoor pure voice intercom solution based on Espressif's products?

For pure voice intercom solutions, the ESP32-S3 series is recommended for development. For application software, refer to the esp-adf/examples/protocols/voip example.

3.3.34 What is the difference between software AEC (Acoustic Echo Cancellation) and hardware AEC in the ESP-ADF audio application development framework?

The difference between hardware AEC and software AEC lies in the source of the reference signal. If the reference signal comes from an external chip (such as ES8311, ES7210) that is independent of the main controller, it is hardware AEC; if the main controller itself replicates the reference signal for echo cancellation, it is software AEC.

3.3.35 Do ESP series products support Voice Activity Detection (VAD)?

Yes, ESP series products support Voice Activity Detection (VAD).

- ESP32, ESP32-C series, ESP32-S series products do not have a dedicated hardware VAD unit, but VAD can be implemented through software algorithms combined with I2S/PDM microphone input. For details, please refer to esp-sr/include/esp32c5/esp_vad.h.
- ESP32-P4 features a built-in hardware VAD unit, which can detect voice activity by configuring the LP I2S and VAD drivers. For more details, please refer to esp-sr/include/esp32p4/esp_vad.h.
- For VAD support on other products, please check under esp-sr/include.

3.4 ESP-BLE-MESH Development Framework

For more FAQs about ESP-BLE-MESH, please refer to ESP-BLE-MESH FAQ.

3.4.1 Why does the ESP-BLE-MESH application wait for a long time during fast provisioning?

After the app provisioned one Proxy node, it will disconnect from the app during fast provisioning, and reconnect with the app when all the nodes are provisioned. The long wait during fast provisioning may be due to:

- Complex network topology. If there are many nodes in the network and the topology is complex, the Provisioner may need more time to scan the network and communicate with the nodes.

- Unstable network signal. If the network signal is unstable, communication may be interfered with or lost, resulting in a longer waiting time for the application.
- Long node response time. If the node response time is long, the Provisioner may wait for a timeout and resend the message, resulting in a longer waiting time for the application.
- Communication failure between the application and the Provisioner. Such failure may result in a longer waiting time for the application.

3.4.2 How to clear the network information of the ESP32 ESP-BLE-MESH node? / How to manually reset the ESP-BLE-MESH device?

You can call esp_ble_mesh_node_local_reset().

3.4.3 How to delete the network information of a node from the provisioning device?

To delete the information of a node, you can call esp_ble_mesh_provisioner_delete_node_with_uuid() or esp_ble_mesh_provisioner_delete_node_with_addr().

3.4.4 When a node is powered off and then powered on again, do I need to re-provision it?

To save yourself from re-provisioning, go to `menuconfig` > `Component config` > `Bluetooth Mesh support` and enable `Store Bluetooth Mesh key and configuration persistently` to save the configuration information.

3.4.5 Assuming development board 1 acts as a Provisioner, and boards 2, 3, and 4 act as nodes. After successful provisioning, if board 1 loses power and is then powered on again, can it join the mesh network?

After board 1 is powered on again, if the NetKey and AppKey have not changed, it can directly join the network. However, the addresses of the nodes in the mesh network will be lost if they are not saved in advance.

3.4.6 In ESP-BLE-MESH, how do I know if a node is offline?

Nodes can periodically publish messages. You can configure them to periodically send Heartbeat messages through the Health Model or vendor messages through the Vendor Model.

3.4.7 How do ESP-BLE-MESH nodes communicate with each other in strings?

They can use the Vendor Model. The sender puts the string into the vendor message, and the receiver parses it as a string after receiving the message.

3.4.8 Failed in initializing the partition when configuring ESP-BLE-MESH to save node information. The error message is `BLE_MESH: Failed to init mesh partition, name ble_mesh, err 261`. How to fix it?

If you selected the `Use a specific NVS partition for BLE Mesh` option, please make sure that the partition.csv file contains a specific partition named `ble_mesh`.

3.4.9 How to add Health Model to the Provisioner's example?

In menuconfig, go to `Component config` > `ESP BLE Mesh Support` > `Support for BLE Mesh Client Models` and enable `Health Client Model`.

3.4.10 What's the difference between using ble_mesh_fast_prov_client as a Provisioner and a mobile phone as a Provisioner?

- The `ble_mesh_fast_prov_server` example configures the model when it receives the ESP_BLE_MESH_MODEL_OP_APP_KEY_ADD opcode, while the phone Provisioner needs to send the ESP_BLE_MESH_MODEL_OP_MODEL_APP_BIND opcode to bind the model's AppKey, and then send the `ESP_BLE_MESH_MODEL_OP_MODEL_PUB_SET` to configure the publication.
- The `ble_mesh_fast_prov_client` and `ble_mesh_fast_prov_server` examples are fast provision solutions provided by Espressif. It can provision 100 nodes within 60 s. To achieve this function, some vendor messages are added for the transmission of vendor information between devices.

3.4.11 Are there any tools or methods to view the encrypted messages transmitted between ESP-BLE-MESH nodes?

- To decrypt the data packet, you need to configure the NetKey, AppKey, DevKey, and IV Index. You can view the configuration interface.
- Broadcast packets should be captured simultaneously on channels 37, 38, and 39, usually using specialized equipment.

3.4.12 Can the AppKey be set by the manufacturer? Is it related to the unicast address?

The application key can be set by the manufacturer. It is bound with the model and is not related to the unicast address.

3.4.13 If a node suddenly goes offline, does the entire mesh network need to poll and send Heartbeat messages through the Health Model monitoring mechanism?

The ESP-BLE-MESH network does not establish any connections. It sends messages directly through the broadcast channel. You can send a heartbeat packet to the same node to check its status.

3.4.14 Can the main node (Proxy node) and the slave node send messages to each other using the client-server model? Is there an example?

Please refer to the ble_mesh_fast_provision/ble_mesh_fast_prov_server example.

3.4.15 In the lower-right corner of the nRF mobile application, tap `Setting` and you will find a configurable `Network Key` field. Whose NetKey does this field configure?

- In the nRF mobile application, the `Network Key` configures the NetKey of the Provisioner. When the Provisioner provisions other devices, this NetKey is assigned to the nodes that are connected to the network.
- If the Provisioner has multiple NetKeys, the Provisioner can choose which NetKey to assign to the device during configuration. The Provisioner can communicate with nodes in the network using different NetKeys. Each node's NetKey is assigned by the Provisioner.

3.4.16 How can a device join the ESP-BLE-MESH network?

You can refer to Getting Started with ESP-BLE-MESH.

3.4.17 What is the maximum packet size for ESP-BLE-MESH transmission?

The maximum single packet size at the application layer is 384 bytes, and the maximum size at the lower layer without packet segmentation is 11 bytes.

3.4.18 Can you provide an application example of provisioning through ESP32 ESP-BLE-MESH? What software can be used as the provisioning configuration application?

- You can use the application example onoff_server and the nRF Mesh mobile application.
- For the provisioning process, refer to Getting Started with ESP-BLE-MESH.

3.4.19 In ESP-BLE-MESH, the default name of the unprovisioned device is ESP-BLE-MESH. How to modify it?

You can use the esp_ble_mesh_set_unprovisioned_device_name() API. It is recommended to call it after esp_ble_mesh_init(). Otherwise, the name will still be the default ESP-BLE-MESH.

3.4.20 How many node devices can the ESP32 ESP-BLE-MESH application connect to?

Theoretically, the ESP32 ESP-BLE-MESH application can connect up to 32767 devices, but the actual number of connected devices depends on memory usage.

3.4.21 After ESP32 runs the ESP-BLE-MESH program for a long time, a segmentation error occurs when the client sends a message to the server, and the ESP-BLE-MESH prints `NO multi-segment message contexts available`. How to solve the issue?

You can go to `Component config` > `ESP BLE Mesh Support` > `Maximum number of simultaneous outgoing segmented messages`, and expand the space by configuring `BLE_MESH_TX_SEG_MSG_COUNT`.

3.4.22 Can I disable NetKey and IV Update when using the ESP32 ESP-BLE-MESH application?

No. NetKey and IV Update must stay enabled.

3.4.23 How to receive other ADV broadcast packets while BLE MESH is enabled?

Enable `BLE_MESH_BLE_COEX_SUPPORT` in menuconfig, register callbacks through esp_ble_mesh_register_ble_callback(), and then start scanning with esp_ble_mesh_start_ble_scanning() to receive other broadcast packets.

3.5 Camera Application

3.5.1 What type of camera does the ESP32 series chip support?

- For camera models supported by the ESP32, ESP32-S2, and ESP32-S3 series, please refer to ESP32 Camera Driver.
- For camera models supported by the ESP32-P4, ESP32-C5, and ESP32-C61 series, please refer to espressif-camera-sensors-component.

3.5.2 What are the output image formats of the camera?

The image format is mainly determined by the camera. If a camera supports multiple image formats, such as RGB565, RGB888, YUV422, JPEG, and so on, the output format then needs to be selected by configuring the camera’ s register.

3.5.3 What parameters does the camera support for adjustment?

The camera supports adjustment of the following parameters: built-in image mode parameters such as image data transfer speed (PCLK), camera output format, resolution, output image size, white balance, GAMMA correction.

3.5.4 What is the relationship between MCLK and PCLK in the camera? What is the difference between them?

- MCLK is the main clock of the camera system, controlling the synchronization and frequency of the system. Inside the camera chip, MCLK is used to control the timing of various modules, such as pre-processors, digital signal processors, pixel arrays, and data output interfaces. Usually, the frequency of MCLK is jointly determined by the system clock of the main control chip and the divider inside the camera. Common frequencies are 6 MHz, 12 MHz, 24 MHz, 48 MHz, etc.
- PCLK is the clock signal used to control pixel output. When the camera outputs an image, each pixel output requires a timing signal, and PCLK is the signal used to control this timing. Specifically, the rising edge of PCLK indicates that the data of a pixel has been output, and the falling edge indicates that the data of the next pixel is about to be output, thus forming a sequence of pixel data.
- MCLK is multiplied/divided (determined by camera configuration) inside the camera to get PCLK. Usually, the frequency of PCLK is half or an integer multiple of half of MCLK. For example, under 24 MHz MCLK, the frequency of PCLK can be 12 MHz, 6 MHz, etc.

3.5.5 Is a higher PCLK always better for the camera?

- Theoretically, the higher the PCLK is, the faster the data transmission will be. However, in actual use, a higher PCLK also means a higher requirement on the processing speed of the chip.
- The current ESP32 and ESP32-S2 chips realize parallel communication through the I2S interface. Too high a PCLK will cause parallel data to be out of sync, resulting in image jitter or even screen flicker.
- ESP32-S3 uses an independent LCD-CAM interface, which can support a higher PCLK frequency.
- For ESP32, the upper limit of PCLK is 8 MHz.
- For ESP32-S2, the upper limit of PCLK is 32 MHz.
- For ESP32-S3, the upper limit of PCLK is 40 MHz.

3.5.6 Do the ESP32 series chips support the MIPI interface?

- ESP32-P4 supports the MIPI interface.

3.5.7 Do the ESP32 series chips support the USB2.0 interface?

- ESP32-S2 and ESP32-S3 support USB2.0 full-speed interface (12 Mbps). In addition, ESP32-P4 also supports USB2.0 high-speed interface (480 Mbps).

3.5.8 Why is the transmission speed of YUV/RGB in the camera slower than JPEG?

- Because the amount of YUV/RGB data is larger than that of JPEG.
- For example: For a screen size of 320 × 240, the output of YUV422 is 153.6 K, while JPEG only needs about 10 K after compression.

3.5.9 In camera applications, what are the factors that may affect the frame rate?

In camera applications, the factors that affect the frame rate mainly include:

- Resolution: As the resolution increases, more pixel data needs to be captured and transmitted per frame, resulting in a lower frame rate.
- Image format: Common image formats include RGB565, RGB888, YUV422, JPEG, etc. Different image formats have differences in image quality and data compression, which will directly affect the frame rate.
- Image processing: If each frame of the image needs to be processed, such as noise reduction, enhancement, compression, etc., it will take up more processing time and reduce the frame rate.
- Transmission bandwidth: As the transmission bandwidth becomes smaller, fewer data needs to be transmitted per frame, resulting in a lower frame rate.
- Processor performance: For the processor with lower performance, it is more difficult to handle the amount of data needed per frame, resulting in a lower frame rate.

Therefore, in camera applications, it is necessary to balance these factors according to specific application scenarios to achieve the best frame rate and image quality.

3.5.10 How to troubleshoot when the camera fails to run?

- Unable to recognize the camera model:
- Check whether the pins correspond correctly, especially for XCLK, SIOC, and SIOD.
- The clock frequency input by XCLK may be too low or the camera power supply is abnormal, causing the camera to fail to run normally.
- There are too many devices mounted on SIOC and SIOD, causing the polled read to return the address ID of a device other than the camera. In this case, it is recommended to fix the camera ID to remove the polling step.
- The camera model is recognized, but there is no image display:
- Check whether there is a signal on the camera data pin, and whether MCLK is input normally.
- Check whether the camera register parameters are configured correctly.
- The camera image display is abnormal:
- Check the code to see whether the output format is RGB, YUV, or JPEG, and whether it meets the format required by the receiving end.
- Try to lower the PCLK frequency.

3.5.11 Does ESP32 support video stream transmission?

- The operation of video stream transmission is divided into binary transmission and video stream encoding and decoding.
- Binary transmission: ESP32 itself supports binary transmission, so whether the video stream transmission is supported depends on the network bandwidth of the transmission. The current ESP32 TCP bandwidth is 20 MB/s, please refer to Wi-Fi test data.
- Video stream encoding and decoding is not yet supported on ESP32.

3.5.12 Where is the factory firmware of ESP-EYE?

Please refer to ESP-EYE’ s factory firmware.

3.5.13 Where are the examples related to the camera solution stored?

- Please refer to ESP-WHO.
- Please refer to esp-iot-solution.
- Please refer to esp-dev-kits.
- For common examples of the ESP32-P4 series, please visit esp-video/examples.
- For examples of using ESP32-P4 together with an LCD screen, please visit esp-iot-solution/examples/camera/video_lcd_display.

3.5.14 Does ESP32 support a camera with a 12-bit DVP interface?

No, the current driver only supports an 8-bit DVP interface.

3.5.15 Can ESP32 use a camera without JEPG encoding to obtain JPEG images?

If the camera itself does not support JPEG encoding, you can refer to our esp-iot-solution/examples/camera/pic_server example, to implement software JPEG encoding on the ESP32 device. This method encodes YUV422 or RGB565 data through software to obtain JPEG images.

3.5.16 Can the 2-megapixel OV2640 camera on ESP-EYE be changed to only output 300,000-pixel images?

Yes, you can specify the resolution size that the camera should output by configuring the frame_size value during initialization.

3.5.17 Does ESP32 support a global shutter camera?

Yes. Currently, the camera models supported by ESP32 are SC031GS, SC035HGS, and OV9281, while other cameras need additional driver support.

3.5.18 What is the frame rate when ESP32 transfers 1080P video via RTSP using the DVP camera?

We have not conducted the test for 1080P yet. Currently, 720P can reach 20 FPS.

3.5.19 ESP32-S3 only supports MJPEG encoding, but H264/H265 format encoding is needed when implementing rtsp/rtmp streaming. Is there any encoding that supports H264/H265 format?

Currently, ESP32-S3 does not support hardware-accelerated H.264/H.265 encoding. However, you can use software encoders, such as the FFmpeg library and the x264/x265 library, to convert MJPEG frames captured from the OV2640 into H.264/H.265 encoded frames. The conversion performance depends on the processor performance, which may affect the frame rate.

3.5.20 Does ESP32/ESP32-S3 support wide-angle cameras?

Yes. You can refer to BF3005 and OV5640.

3.5.21 It takes five seconds for ESP32-S2 to display the camera image from power-on. Is there room for improvement?

Yes, please refer to the following:

- Try to remove some delay functions in `esp_camera_init()`.
- Change the sccb clock frequency in `menuconfig` > `component config` > `camera configuration` to 400000.

3.5.22 Can ESP32 directly support 24 MHz frequency to the GC0308 camera?

It might not be feasible. According to tests, the maximum stable test value of XCLK ESP32 supported for GC0308 is 20 MHz.

3.5.23 Does ESP32/ESP32-S3 support the MMS streaming protocol?

No, ESP32 and ESP32-S3 do not support the Microsoft Media Server (MMS) streaming protocol directly. MMS is a streaming media transmission protocol developed by Microsoft, mainly used for network streaming media playback in Windows Media Player. The streaming media protocols supported by ESP32 and ESP32-S3 are RTSP and SIP. If you need to use ESP32 or ESP32-S3 for scenarios that require MMS protocol support, you may consider using middleware or converters that support the MMS protocol.

3.5.24 When debugging the GC2145 camera with ESP32-S3, the maximum supported resolution seems to be 1024x768. If it is adjusted to a larger resolution, such as 1280x720, it will print cam_hal: EV-EOF-OVF error. How to solve this issue?

In this case, it is necessary to reduce the PCLK of GC2145. For specific methods, try to configure a smaller XCLK and debug the PLL clock coefficient of the camera.

3.5.25 Does ESP32-S3 support the GB28181 protocol?

ESP32-S3 does not directly support the GB28181 protocol, but it can be implemented by combining ESP32-S3 with external circuits and software. Since GB28181 is a communication protocol between video surveillance devices, the network capabilities of ESP32-S3 and external circuits, such as video encoders, audio codecs, and sensors, can be used to implement the GB28181 function. At the same time, relevant software development is required to realize the parsing and data transmission of the GB28181 protocol.

3.5.26 Is there any reference for ESP32/ESP32-S2/ESP32-S3 to recognize the QR code through the camera?

Yes, please refer to the code recognition in ESP-WHO.

3.5.27 When adding the SD-card interface and camera interface for OV5640 sensor, we found that some pins of different ESP32 drivers conflicted with each other. Please suggest pins for the camera interface and SD-card interface.

The ESP-WROVER-KIT development board includes the camera and SD card circuits, so you can refer to pins configuration of the ESP-WROVER-KIT V3 getting started guide.

3.5.28 Can a driver for a specific camera model be added if the currently supported camera sensors do not meet my requirements?

Yes. Please confirm your requirements and select the camera sensor model with our engineers through technical support. We can then provide the corresponding driver for your camera sensor.

3.5.29 How to add a custom resolution?

Suppose you need a resolution of 640x240, you can use the custom resolution in the following two ways: - Configure the sensor to work at the typical resolution of 640x480, and then only use the upper half of the data (640x240). - Add the identifier FRAMESIZE_640*240 in esp32-camera/driver/include/sensor.h, and define the length and width of that resolution in esp32-camera/driver/sensor.c as {640, 240, ASPECT_RATIO_16X9}. This method requires support for custom resolutions in the sensor’ s driver to work properly.

3.5.30 How to modify the register configuration of the camera sensor?

Suppose you need to change the register configuration of the OV5640 sensor. This can be achieved in two ways: - Directly configure the relevant registers using write_reg() in the reset() function of esp32-camera/sensors/ov5640.c. - Configure the relevant registers at the application layer through the set_reg() function:

```
// Initialize the camera
esp_err_t ret = esp_camera_init(&camera_config);
sensor_t *s = esp_camera_sensor_get();
s->set_reg(s, 0xFFFA, 0xFF, 0xA1);
```

3.5.31 What is the reason for triggering "cam_hal: EV-VSYNC-OVF" in esp32-camera?

This issue occurs when the frame synchronization signal triggered by the sensor is too fast. You can troubleshoot it following the steps below: - Run the esp-iot-solution/examples/camera/pic_server example. If this example runs normally, it indicates that the issue is not hardware-related. - Check the XCLK and resolution specified during sensor initialization. A smaller resolution or a larger XCLK can cause the frame synchronization signal triggered by the sensor to be too fast. Note that the XCLK used by the sensor should match the specified resolution.

3.5.32 What could be the reason for the following warning log appearing in the Camera application based on ESP32-S3?

```
W (7232) cam_hal:FB-OVF
W (7242) cam_hal:FB-OVF
W (7492) cam_hal:FB-OVF
W (7512) cam_hal:FB-OVF
W (7762) cam_hal:FB-OVF
W (7772) cam_hal:FB-OVF
W (8022) cam_hal:FB-OVF
W (8042) cam_hal:FB-OVF
```

The above warning log indicates a frame buffer overflow, which is caused by too fast a frame rate. You can try to reduce the XCLK (Note that the XCLK of ESP32S3 is divided from the 80 MHz clock by default, so the size of XCLK must be divisible by 80 MHz). Specifically, if the sensor is operating in JPEG mode, you can try to increase the size of the jpeg recv buffer by increasing the value of the *Custom JPEG mode frame size (bytes)* option in menuconfig.

3.5.33 What is the difference between the two capture modes of the ESP32-Camera?

After initialization, the camera sensor pushes image data to the receiver on the ESP32.

- When the configured receive mode is CAMERA_GRAB_WHEN_EMPTY, the background driver writes image data to the frame_buffer as long as there is an idle frame_buffer. When all the frame_buffers are exhausted, the new image data pushed by the camera sensor will be forcibly discarded due to the lack of available frame_buffer.
- When the configured receive mode is CAMERA_GRAB_LATEST, the number of frame_buffers that the application layer can obtain is fb_count - 1. This is because the background driver occupies one frame_buffer and tries to refresh the latest data into this frame_buffer.

Note that the capturing does not occur when calling *esp_camera_fb_get*. The capturing is an ongoing process, and we can only control the frame_buffer used by the backend to obtain new data. Therefore, if you want to immediately obtain a new image, try executing the following code:

```
// Returns a frame_buffer to the backend driver
esp_err_t ret = esp_camera_fb_return(esp_camera_fb_get());
// The background program automatically refreshes the new image data to␣
↪frame_buffer, then the application layer can access the data in frame_
↪buffer.
fb = esp_camera_fb_get();
```

3.5.34 How to implement frame skipping with the esp32-camera SDK?

You can call *esp_camera_fb_return(esp_camera_fb_get());* to discard the current frame, that is, to skip the frame that is being fetched.

3.5.35 Can ESP32-S3 connect to two cameras and display split screen?

ESP32-S3 can connect to two cameras with SPI interfaces (with a relatively small resolution, 240*320). However, if using the DVP interface, multiple cameras cannot be used simultaneously. In such cases, the ESP32-P4 is a more suitable option.

3.5.36 Does ESP32-S3 support a 10-bit DVP camera?

In general, a 10-bit DVP camera can be used by capturing only the upper 8 bits, which is sufficient to produce a normal image.

3.5.37 How to scale image data?

- For JPEG images, use the esp_new_jpeg component to directly scale down the decoded data during the decoding process.
- For RGB or YUV data, use the PPA peripheral (only supported on P series chips), or use the software-based image processing component esp_image_effects for scaling.

3.5.38 What' s the difference between the two camera application components esp32-camera and esp-video?

- The chips supported by esp32-camera include ESP32, ESP32-S2, ESP32-S3. In addition, this component only supports camera sensors using the DVP interface.
- The chips supported by esp-video include ESP32-P4, ESP32-S3, and the ESP32-C series chips. This component supports camera sensors with interfaces including SPI, DVP, USB, and MIPI-CSI.

3.5.39 How to add a new camera driver for applications using the esp-video framework?

- Please refer to the document add_new_camera_sensor_driver.

3.5.40 Can I connect multiple cameras to ESP32-P4?

Sure. The DVP and MIPI-CSI interfaces of ESP32-P4 can each connect to one camera. The USB and SPI interfaces can each connect to multiple cameras.

3.5.41 Is it necessary to pair a camera sensor that can only output RAW format data with an ISP module to output clear images?

Yes. RAW data is unprocessed raw data, which requires the ISP module to perform operations such as noise reduction, color restoration, and automatic exposure control.

3.5.42 What equipment is needed to debug the ISP module?

Debugging the ISP module requires standard light sources, colorimeters, 24-color cards, and other equipment.

- For projects that have passed evaluation, you can contact Espressif for assistance in debugging ISP module parameters.
- For camera sensors that have already been debugged, you can fine-tune the image effects by referring to the ISP image processing algorithm document Espressif Image Process Algorithm for ISP.

3.5.43 How does esp-video-components output XVCLK to the camera sensor?

XVCLK/XCLK/MCLK is the master input clock for the camera sensor. It can be supplied by an external crystal oscillator or from an MCU pin. See the esp_cam_sensor xclk_generator test example.

3.6 ESP Matter

3.6.1 Which ESP modules can be connected to Matter?

- Please refer to the Espressif Matter Platform.

3.6.2 What learning materials are available to get started with ESP Matter?

You can refer to:

- ESP Matter Github
- Espressif Matter series blog
- Espressif Matter solution video
- Espressif Matter Demo video

3.6.3 Can I compile and develop ESP Matter on Windows?

- It is currently not possible to compile on Windows. Also, it is not recommended to use a virtual machine for ESP Matter development, as the Matter Controller will use Bluetooth hardware where unknown errors may occur with the virtual machine. Using Linux or macOS systems directly are recommended.

3.6.4 Where can I find the Matter protocol documentation?

Currently, Matter’ s standard protocol has been made public, you can submit a request and download the documentation on the CSA Alliance official website.

3.6.5 How to register an ESP Matter product?

- To register a Matter product requires a CSA membership. After the product passes the Matter certification test, it can be registered with the CSA.

3.6.6 When using Ubuntu virtual machine to develop ESP Matter on ESP32-C3, network provisioning failed when I followed the Matter official tutorial. What could be the reason?

It is not recommended to use a virtual machine for ESP Matter development, as the Matter Controller will use Bluetooth hardware where unknown errors may occur with the virtual machine. It is recommended to use Ubuntu 20.04 LTS and above hosts directly.

3.6.7 How to apply for a Matter DAC?

There are three ways to apply for a Matter DAC:

- Cooperate with an established PKI provider: Many organizations already have a certificate authority that they rely on to obtain their public key infrastructure certificates. In some cases, a Device Attestation Certificate (DAC) can be obtained from such a PKI provider.
- Use your own PKI: Many organizations already have public key infrastructures for code signing, existing device authentication requirements, or other tasks that rely on asymmetric encryption. And public keys are distributed via digital certificates.
- Cooperate with your platform vendor (Espressif): The platform vendor may embed a DAC in a chip or platform module using their VID/PID. CD is used to remap VID/PID using the dac_origin_vid/dac_origin_pid fields.

3.6.8 Does ESP Matter have a test tool/test app?

- Yes, it is recommended to use chip-tool.

3.6.9 Matter needs to use DCL in the network provision process. What is the specific function of DCL?

- Matter DCL is a secure and encrypted distributed storage network based on blockchain technology, which allows the Connection Standards Alliance (CSA) and authorized suppliers to publish their Matter device information. It also allows the Matter ecosystem to query related information through DCL clients.
- For simplicity, Matter DCL will be used for device verification and OTA.

3.6.10 How to connect our Zigbee-based products with Matter through ESP chip?

- The device based on ZigBee technology is not a Matter standard device. At this time, you need to bridge the ZigBee device through the Matter Bridge device to access the Matter network.
- Matter Bridge devices can be implemented using an Espressif Wi-Fi chip + 802.15.4 chip. Matter Bridge For BLE Mesh devices can be implemented with one Espressif Wi-Fi chip + BLE chip, or only one Wi-Fi + BLE combo chip.

3.6.11 Does Matter work with Samsung’ s smartthings?

- Yes, please refer to Configuration test smartthings official blog.

3.6.12 Can Matter-enabled ESP devices be remotely controlled using Amazon/Google/Apple voice devices? Do these voice devices need to support the Matter protocol? (For example: Saying “Turn off the light” to turn off the lights in the house)

- Using Amazon/Google/Apple voice devices that support Matter protocol, it is possible to remotely control Mattter ESP devices. In addition, if other ecosystems also support the ecosystem of the Matter protocol, then the home hub devices such as speakers in this ecosystem can also control the Matter devices remotely.
- The specific practical steps are: build a Matter application scenario for testing based on the esp-matter SDK. - Google Matter Test Method - Apple Matter Test Method

---

3.6.13 Does the product need to pass WiFi authentication and Bluetooth BQB authentication before submitting the Matter authentication?

- Yes. Matter is a protocol that runs on other technologies such as Wi-Fi, Ethernet, Thread, and Bluetooth. Before the Matter authentication, the device must be pass the transport layer protocol authenticated. This requires not only the original Wi-Fi or Thread authentication, but also the BQB authentication of the Bluetooth SIG, given that Matter requires the use of Bluetooth for provisioning.

3.6.14 Where is the DAC (Device Attestation Certificate) pre-imported by ESP Matter module stored?

- The DAC (Device Attestation Certificate) pre-imported by the ESP Matter module is stored in flash. In the Matter Pre-Provisioning service, the Matter DAC certificate is pre-flashed in the esp_secure_cert partition. An example of adding this partition to a partition table is as follows:

```
# ESP-IDF Partition Table
# Name,          Type, SubType, Offset,  Size, Flags
esp_secure_cert, 0x3F,    ,     0xd000,  0x2000,  , # Never mark this as␣
↪an encrypted partition
```

3.6.15 Can I configure Wi-Fi of ESP32 Matter devices by BLE ?

- ESP32 Matter devices can be provisioned via BLE, all application test examples under our esp-matter SDK are configured through BLE. Please refer to section 2.2 Commissioning and Control for instructions.

3.6.16 How to solve the following error when provisioning the device with the command `pairing ble-wifi` or `pairing code-wifi` using the light example in esp-matter via chiptool?

- The error log is as follows:

```
[1707138771.417762][46462:46464] CHIP:TOO: Pairing Failure: src/platform/Linux/
↪BLEManagerImpl.cpp:664: CHIP Error 0x00000003: Incorrect state
[1707138771.417789][46462:46464] CHIP:TOO: Run command failure: src/platform/
↪Linux/BLEManagerImpl.cpp:664: CHIP Error 0x00000003: Incorrect state
[1707138771.417820][46462:46464] CHIP:BLE: No adapter available for new␣
↪connection establishment
```

- The above error is due to the computer not having a Bluetooth adapter. Please use a computer with a Bluetooth adapter. Laptops generally come with a Bluetooth adapter. If you are using a desktop computer, you will need a Bluetooth dongle to provide Bluetooth functionality.

3.6.17 How to generate and install Matter QR codes during mass production?

You can use the esp-matter-mfg-tools tool to generate factory information for devices in batch. Then on the production line, flash the generated information and attach the corresponding QR code labels for each device. At the same time, record the mapping between each device ' s MAC address and its factory firmware for future traceability.

3.7 ESP-NOW

3.7.1 What is the one-to-one bit rate for ESP32 in ESP-NOW mode ?

Test result:

- Test board: ESP32-DevKitC V4.
- Wi-Fi mode: station.
- PHY rate is 1 Mbps by default.
- Around 214 Kbps in opened environment.
- Around 555 Kbps in shielding box.
- If you require a higher rate, it ' s feasible to configure the rate through esp_wifi_config_espnow_rate.

3.7.2 What is ESP-NOW? What are its advantages and application scenarios?

- ESP-NOW is a connectionless communication protocol defined by Espressif.
- In ESP-NOW, application data is encapsulated in action frames from different vendors and then transmitted from one Wi-Fi device to another without a connection.
- ESP-NOW is ideal for smart lights, remote control devices, sensors, and other applications.

3.7.3 Can Wi-Fi be used with ESP-NOW at the same time?

- Yes, but it should be noted that the channel of ESP-NOW must be the same as that of the connected AP.

3.7.4 How do I set the rate at which ESP-NOW data is sent?

You may use the esp_wifi_config_espnow_rate() function to configure the rate, such as `esp_wifi_config_espnow_rate(WIFI_IF_STA, WIFI_ PHY_RATE_MCS0_LGI)`.

3.7.5 ESP-NOW allows pairing with a maximum of 20 devices. Is there a way to control more devices?

You can use broadcast packets and provide the destination addresses in the payload. The number of addresses would then not be affected by the limited number. You only need to configure the correct broadcast address.

3.7.6 What is the maximum number of devices that can be controlled by ESP-NOW?

This depends on the specific communication method:

- If unicast packets are used, up to 20 devices can be paired and controlled at the same time.
- If ESP-NOW encrypted mode is used, up to 6 devices can be paired and controlled at the same time.
- If broadcast packets are used, theoretically there is no limitation in the number of devices that can be controlled. You only need to configure the correct broadcast address and take care of the interference issue when too many devices are paired.

3.7.7 Do I need to connect a router for communication between ESP-NOW devices?

ESP-NOW interacts directly from device to device and does not require a router to forward data.

3.7.8 Why does ESP-NOW limit the data length of each packet to 250 bytes? Can it be configured?

- The maximum length does not support configuration. ESP-NOW uses one vendor-specific element field of action frame to transmit ESP-NOW data, whose length field is only 1 byte (0xff = 255) as defined by IEEE 802.11. Thus, the maximum length of ESP-NOW data is limited to 250 bytes.
- Alternatively, you may try with API `esp_wifi_80211_tx()` to send and use sniffer mode to receive. This way could fulfill the need of working only base on Wi-Fi stack without involving TCP/IP stack.

3.7.9 What should I pay attention to when using ESP-NOW applications?

- The device cannot switch channels after connecting to Wi-Fi. It can only transmit and receive data on the current Wi-Fi channel.
- After the device enters Modem-sleep mode, it cannot receive data from ESP-NOW.

3.7.10 How can I reduce power consumption when using ESP-NOW?

- You can use the following methods to reduce power consumption:
  - If you use ESP-IDF in versions earlier than v5.0, when the AP is not connected, you can configure the wake-up window size and interval using the esp_now_set_wake_window() and esp_wifi_set_connectionless_wake_interval() functions respectively to save power.
  - If you use ESP-IDF v5.0 or the latest master version, the functions are different from the other versions. Whether the AP is connected or not, you can use the esp_now_set_wake_window() and esp_wifi_connectionless_module_set_wake_interval() functions to set the wake-up window size and interval, respectively.
- Note that the issue of window synchronization between the sending end and receiving end needs to be considered in the application layer design. In this way, the chip will wake up at every "interval" and work for a period of time equalling the value of "window size". Under this situation, you also need to configure CONFIG_ESP_WIFI_STA_DISCONNECTED_PM_ENABLE=y in sdkconfig.defaults.

3.7.11 In addition to wireless communication through ESP-NOW, is there any other better way to realize one-to-one and one-to-many communication?

One-to-one and one-to-many communication can also be realized by using SoftAP + Station. The master device applies Wi-Fi SoftAP mode to establish connections with multiple slave devices (Wi-Fi Station) at the same time.

3.7.12 Do ESP-NOW applications support sending packets over each Wi-Fi channel?

Yes, please refer to ESP-NOW documentation.

3.7.13 Are there any special procedures required if I intend to use ESP-NOW for commercial purposes? Could you provide technical documentation on ESP-NOW? To evaluate the quality of wireless communication, I would like to know relevant information about parameters including CSMA/CA, modulation method, and bit rate.

- The application for ESP-NOW does not require any special procedures.
- For technical documentation, please refer to ESP-NOW User Guide. You can use examples in ESP-NOW SDK for testing.
- The default bit rate of ESP-NOW is 1 Mbps.

3.7.14 I tested the application esp-idf/examples/wifi/espnow using ESP32. Does it only support connecting to 7 encrypted devices at the maximum?

- In the esp-now application, ESP32 supports connecting to no more than 17 encrypted devices, and the default value is 7. For more details, please refer to "Add Paired Devices" .
- If you want to change the number of paired encryption devices, set `CONFIG_ESP_WIFI_ESPNOW_MAX_ENCRYPT_NUM` in WiFi component configuration menu.

3.7.15 How do I obtain the corresponding RSSI when transferring data using ESP-NOW?

- You can use wifi_pkt_rx_ctrl_t to get the corresponding RSSI.

3.7.16 How do I use RSSI in ESP-NOW to achieve selective range control?

- You can achieve it by modifying the espnow_frame_head_t g_initiator_frame() function with .forward_ttl and .forward_rssi parameters. For corresponding parameter descriptions, please see espnow/src/espnow/include/espnow.h.

3.8 ESP RainMaker cloud service

3.9 Third-Party Cloud Service

3.9.1 Are there any examples for OTA upgrading?

Please refer to the following links:

- For ESP8266, please refer to ESP8266 OTA.
- For ESP32 series, please refer to ESP32 series OTA.

3.9.2 How to integrate ESP32 with Tmall Genie? Any reference materials?

You can integrate ESP32 with Tmall Genie using the `esp-aliyun` SDK with reference to Guide for Integrating ESP Devices with Alibaba Cloud.

3.9.3 What is the difference between `esp-aliyun` and `esp-ali-smartliving`?

- `esp-aliyun` is used for integration with "Internet of Things Platform".
- `esp-ali-smartliving` is used for integration with "Smart Living IoT Platform".
- Alibaba Cloud ensures the two platforms are interconnected in the cloud. They are functionally similar and can be used interchangeably.
- For the differences between the two, please refer to The differences between the Smart Living IoT Platform and the IoT Platform.

3.9.4 How to ask questions and provide feedback when encountering issues in connecting Espressif products to the Espressif cloud platform?

- For ESP RainMaker, the cloud platform developed by Espressif, submit feedback to GitHub.
- For other cloud platforms, please send the information about the specific platform along with your issues to sales@espressif.com.

3.9.5 Can ESP32 and ESP8266 connect to Alexa or Google Home?

- Yes. For how to connect to Alexa, refer to the aws_iot example and make some configuration.
- There are currently no examples for Google Home, but you can refer to the ESP32 application example in esp-google-iot.

3.9.6 What should I do to connect ESP32 to Alibaba Cloud via Ethernet and MQTT?

Use esp-aliyun but replace the Wi-Fi initialization code with Ethernet initialization. You can refer to the Ethernet examples under ESP-IDF.

3.9.7 How to generate triplet bin files in bulk on Alibaba Cloud?

The instructions in the document have been updated. On version 4.4.X, please use the following command to generate the corresponding bin files: *python $IDF_PATH/tools/mass_mfg/mfg_gen.py generate multipule_mfg_config.csv multipule_mfg_values.csv scgw 0x4000*.

3.10 ESP-WIFI-MESH Application Framework

Note: If you have new requirements for Wi-Fi Mesh applications, it is recommended to use our newly released ESP-Mesh-Lite solution, instead of Wi-Fi Mesh.

3.10.1 How much memory does ESP-WIFI-MESH occupy? Is external PSRAM required?

ESP-WIFI-MESH occupies about 60 KB of memory. Whether external PSRAM is needed depends on the complexity of the application scenario. Generally, external PSRAM is not required for typical applications.

3.10.2 Can ESP-WIFI-MESH support performing OTA for multiple devices at a time?

ESP-WIFI-MESH devices support performing OTA for multiple devices at a time. The OTA approach is that the root node downloads the firmware and then disseminates it to other nodes. For a specific example, please refer to Mupgrade.

3.10.3 How many devices can ESP32 support for ESP-WIFI-MESH networking?

ESP32 can support 1,000 devices for ESP-WIFI-MESH networking. If you want to stably connect a large number of devices, it is recommended to avoid connecting over 512 networking devices under each ESP-WIFI-MESH network.

3.10.4 What is the difference between the ESP-WIFI-MESH Router mode and the No Router mode of ESP32?

- The Router mode refers to router-based networking, in which the root node connects to the router.
- The No Router mode refers to self-networking without a router, in which external data interaction is not possible.

3.10.5 Can ESP-WIFI-MESH of ESP32 complete networking when the child node fails to detect the router signal?

With the same configuration SSID, even if the child node fails to detect Wi-Fi, ESP-WIFI-MESH of ESP32 supports connecting the child node to the root node.

3.10.6 Can ESP-WIFI-MESH of ESP32 automatically repair the network?

ESP-WIFI-MESH of ESP32 has a mechanism to detect network disconnection and can automatically repair the network.

3.10.7 When using ESP-WIFI-MESH of ESP32 without connecting to Wi-Fi, how can I realize a self-organizing network?

To designate a device as the root node, please refer to the Mwifi description and example.

3.10.8 When using ESP-WIFI-MESH of ESP32 for automatic root node election during networking, is it possible to designate specific local modules for the election process?

ESP-WIFI-MESH of ESP32 supports designating a device as a child node. Once designated, the device will not participate in the root node election, thereby realizing a localized Mesh network for the election.

3.10.9 Can multiple root nodes send messages to each other in a no-router scenario using ESP-WIFI-MESH of ESP32?

In a no-router scenario, multiple root nodes cannot send messages to each other.

3.10.10 How to query the source code of the ESP-WIFI-MESH application?

- iOS source code: https://github.com/EspressifApp/EspMeshForiOS
- Android source code: https://github.com/EspressifApp/EspMeshForAndroid

---

3.10.11 Does ESP-WIFI-MESH offer a no-router solution for self-networking?

Please refer to the No Router and get_started examples for no-router solutions.

3.10.12 After using Mwifi for automatic networking, how can I obtain the signal strength (RSSI) of all potential parent nodes of a specific node?

- You can call `mwifi_get_parent_rssi()` to obtain the signal strength of the parent node.
- You can refer to the ESP-NOW Debug Receiver Board Demo for ways to obtain the signal strength of other nodes.

3.10.13 What protocol is used for communication between nodes in the ESP-MDF Mesh network?

Communication within the Mesh network is based on a custom protocol at the data link layer, which is one of our core protocols. It supports the ACK mechanism but lacks a built-in timeout/retransmission mechanism. You can add the corresponding mechanism at the application layer according to your needs.

3.10.14 Can ESP-WIFI-MESH connect all nodes to the router?

Only the root node can connect to the router. The child nodes will connect directly or indirectly to the root node, and then communicate with the router through the root node.

3.10.15 Can the root node of ESP-WIFI-MESH connect to the Internet via 4G dial-up?

It is possible, but there is currently no dedicated application for this scenario. You can refer to the No Router example in ESP-MDF, where the root node communicates directly with the computer via the serial port, or it can be configured to transmit data via a 4G module.

3.10.16 After a successful connection using the esp_mesh_set_parent function, if the AP is disconnected, the function will continuously attempt to reconnect. How can I set the number of reconnect attempts?

- For the self-networking solution, ESP-WIFI-MESH will not attempt to reconnect by default. If the AP is disconnected, you need to call `esp_wifi_scan_start` to get the devices that can be connected and reset the parent node. Please refer to Mesh Manual Networking Example.
- It is recommended to use the self-networking solution for development.

3.10.17 Why is `phy_init: failed to load RF calibration data` reported after setting the button?

The Espressif chip will self-calibrate RF on the first power-up and store the data in NVS. If this part is erased, this error will be reported and a full calibration will be performed.

3.10.18 How to pause/resume Mwifi?

Call `mwifi_stop/mwifi_start` to pause/resume Mesh.

3.10.19 For the ESP32 series without router Mesh networking, how can an application connect to the softAP of the root interface?

In the scenario without a router Mesh, the softAP initiated by Mesh devices does not support the connection of devices other than Mesh devices. If there is a need for non-Mesh devices connection, such as smartphones, it is recommended to use ESP-Mesh-Lite.

3.10.20 ESP-WIFI-MESH can connect to an AP but cannot connect to the TCP server on that AP. How can this issue be resolved?

Please refer to the GitHub issue: mesh -> "with-router" example doesn't work with espressif IDF softAP #71.

3.10.21 How to modify the AP connection and maximum layer in the Mwifi example? What is the maximum bandwidth and delay during communication?

- You can go to menuconfig and modify the configuration through `Component config > MDF Mwifi > Capacity config`.
- For communication performance, please refer to the Performance section.
- WIFI-MESH bandwidth can be tested through the ESP-WIFI-MESH Console Debugging Demo.

3.10.22 How to get real-time return values of the sensor?

As the device end is functioning as an HTTP server, requests must be initiated by the application to obtain real-time data. You can employ the following two methods to acquire real-time data:

- When the sensor data changes, notify the mobile phone to actively request data through UDP. If you are using our local communication protocol, send the following command to prompt the application to request device data:

```
data_type.protocol = MLINK_PROTO_NOTICE;
ret = mwifi_write(NULL, &data_type, "status", strlen("status"), true);
MDF_ERROR_CONTINUE(ret != MDF_OK, "<%s> mlink_handle", mdf_err_to_
↪name(ret));
```

- Establish a TCP/MQTT/HTTP server. Once a TCP connection is established with the server, sensor data changes will be actively reported.

3.10.23 For a new node that might already be installed in a device that is located at a considerable distance from the root node, how can this node join the ESP-WIFI-MESH network?

- It seems that you are using the get-started example. For testing convenience, this example is a no-router solution, in which the root node is designated. Therefore, if the root node crashes, the other devices will not be able to recover.
- You can refer to the light example in development_kit. This example can be used with the ESP-Mesh application (the Android version can be downloaded from our official website, and the iOS version can be downloaded and tested by searching for ESP-Mesh in the App Store).
- This example does not designate a root node. Instead, the root node is elected by the devices, which requires a router. In this scenario, if the root node fails, the remaining devices will automatically complete the re-networking and connect to the router without user intervention.

3.10.24 Is the source code for the ESP-WIFI-MESH application open?

- We have already opened the source code for the ESP-Mesh application on GitHub. Please refer to EspMeshForAndroid and EspMeshForiOS.
- If you have any questions or encounter bugs during usage, feel free to leave comments and ask questions on GitHub or here. We will handle them as soon as possible.

3.10.25 What is the maximum packet size for Wi-Fi Mesh data transmission?

The maximum packet size is 1,456 bytes.

3.10.26 Does ESP32' s Wi-Fi Mesh support self-networking without a router?

Yes, please refer to the No Router example.

3.10.27 What is the maximum number of node layers allowed when using ESP32 with Wi-Fi Mesh?

- In a Wi-Fi Mesh network, you can set the maximum number of network layers using the esp_mesh_set_max_layer() function.
- For a tree topology, the maximum number is 25. For a chain topology, the maximum number is 1,000.

3.10.28 When testing the esp-mdf/examples/function_demo/mwifi/router example with the ESP32 development board, the device name displayed on the router connection end is "espressif" after the ESP32 connects to the router. How can I change this name?

You can go to menuconfig, and modify it through `Component config` > `LWIP` > `(espressif) Local netif hostname`.

3.10.29 Can Wi-Fi Mesh send messages to specific nodes through a TCP server?

Wi-Fi Mesh can send data to specific nodes or group addresses through a TCP server. Please refer to the Mwifi Router example.

3.10.30 During the operation of the ESP32 Wi-Fi Mesh network, what event does the system feedback if the root node is lost?

If the root node is lost, all nodes will trigger the MDF_EVENT_MWIFI_PARENT_DISCONNECTED (MESH_EVENT_PARENT_DISCONNECTED) event. Subsequently, the nodes will commence a re-scan and initiate a new election process until a new root node is elected.

3.10.31 When developing Wi-Fi Mesh applications with ESP32 using the esp_mesh_send() function, I' ve observed that the server is not receiving any data. How can data be transmitted from leaf nodes to an external server?

- esp_mesh_send() can only be used for internal data communication in the Wi-Fi Mesh network.
- To send data from a leaf node to an external server, the data needs to be forwarded through the root node.
- The correct approach is for leaf nodes to first send the data to the root node, and then the root node forwards the data to the external server.

3.10.32 How to perform an OTA upgrade after ESP-MESH devices are networked?

- The root node can connect to the server to get the upgrade bin file, and then send the firmware to the corresponding module for OTA upgrade via MAC address.
- For details, please refer to the Mupgrade Example.

3.10.33 Is there a reference design for ESP-MESH lights?

- The overall design of the light is completed by a third-party factory, and we do not have related schematics or PCB layouts. But from the module perspective, you only need to power the chip, and the chip outputs PWM to control the color or color temperature change of the light, which is not too complicated.
- Please refer to ESP-MDF for more information about Mesh.

3.10.34 What is the default mode of ESP-MESH nodes?

The default mode of ESP-MESH is IDLE.

3.10.35 When ESP-MESH starts in AP+STA mode, can the phone search for the AP?

No, as ESP-MESH is a private protocol of Espressif. For details, please refer to the WIFI-MESH Introduction.

3.10.36 After the device has completed networking, is it necessary to perform a complete re-scan when adding a new device?

No, it is not necessary. Only a scan within the current set of child nodes is required. The new device can connect to the node with the strongest signal strength as its parent node.

3.10.37 Can ESP32, as the master device, achieve time synchronization with multiple slave devices with an error within 2 ms?

- For this application scenario, it is recommended to develop based on ESP-MDF. Please refer to light_example.
- Using esp_mesh_get_tsf_time() can meet the precision requirements mentioned (within 2 ms).

3.10.38 How to get the node type in ESP-MESH?

You can call the esp_mesh_get_type interface to get the node type.

3.10.39 Is there an example of ESP-Mesh root node sending messages to the server via Ethernet?

Please refer to the Mwifi Router Example.

3.10.40 Does the ESP-Mesh-Lite solution support applications without a router?

- Yes, it does. The application scenarios supported by the ESP-Mesh-Lite solution can be found in the ESP-Mesh-Lite Features.
- You can enable the `Component config > ESP Wi-Fi Mesh Lite > Enable Mesh-Lite > Mesh-Lite info configuration > [*] Join Mesh no matter whether the node is connected to router` configuration option for testing based on the Mesh Local Control Example.
- For solutions without a router, please note the following:
  - Try to determine a root node, which can be set through `esp_mesh_lite_set_allow_level(1)`.
  - For other nodes, it is recommended to use the `esp_mesh_lite_set_disallow_level(1)` function to prevent them from becoming root nodes.
  - In the application scenario of Mesh-Lite, establishing a Mesh network relies on factors such as the physical distance between devices and Wi-Fi signal quality. Therefore, thorough on-site testing and debugging are necessary to ensure the performance and stability of the Mesh network.

3.10.41 When ESP-WIFI-MESH is networked, is it possible for the root node or child nodes to simultaneously initiate a Wi-Fi Scan to scan for available AP information around?

No. When ESP-WIFI-MESH is networked, none of the node devices support the activation of the Wi-Fi Scan function.

3.10.42 When using the ESP-WIFI-MESH Router solution, how to switch to a new router for networking?

- You can modify the following code after the `MESH_EVENT_PARENT_DISCONNECTED` event:

```
mesh_router_t change_router = {
    .ssid = "TP-LINK_CSW",
    .password = "12345678",
    .ssid_len = strlen("TP-LINK_CSW"),
};
esp_mesh_set_self_organized(false, false);
esp_mesh_set_router(&change_router);
esp_mesh_set_self_organized(true, true);
```

3.10.43 Does the ESP-MESH-LITE application support Wi-Fi Power Save mode?

No. In ESP-Mesh-Lite, power consumption can be reduced by increasing the beacon interval or decreasing the Wi-Fi TX power.

3.10.44 Both devices are root nodes connected to the router but the mesh network is not formed. How to fix this?

A common cause is inconsistent Mesh IDs between the two devices. Mesh-Lite requires the same Mesh ID and router credentials (SSID and password) to form one network.

3.10.45 When using the Mesh-Lite App for provisioning, the device is connected to Wi-Fi 1 and then switched to Wi-Fi 2, but after `esp_restart` it still connects to Wi-Fi 1. How to fix this?

Call `esp_mesh_lite_erase_rtc_store` before `esp_restart` to clear the mesh networking information stored in RTC.

3.10.46 Can Mesh-Lite child nodes directly maintain an MQTT long connection?

Yes, there's no need to send the data to the root node first; the child nodes can directly maintain MQTT long connections.

Chapter 4

Software framework

4.1 Bluetooth

4.1.1 Bluetooth LE & Bluetooth

When porting example gatt_server, an error occurred indicating head file does not exist. What could be the reasons?

When porting example gatt_server, an error occurred as `fatal error: esp_gap_ble_api.h: No such file or directory`, but this file is already included in the head file.

- Check sdkconfig to see whether sdkconfig.defaults is ported from the example or not. By default, Bluetooth® is disabled in SDK and needs to be enabled manually.
- If you are using cmake, the link configurations in the CMakeLists.txt file should be copied from the example too.

Does ESP32 support Bluetooth 5.0?

No, the ESP32 hardware only supports Bluetooth LE 4.2.

ESP32 has passed Bluetooth LE 5.0 certification, but does not support the new functions of Bluetooth LE 5.0. If you need to use the features of Bluetooth LE 5.0, please choose other ESP32 series chips.

For Bluetooth certification information of the ESP32 series chips, please refer to ESP32 Bluetooth LE Certification Information.

After the Bluetooth® LE starts advertising, why some mobile phones cannot successfully scan them?

- Please check whether your mobile phone supports Bluetooth LE function. Some mobile phones, such as iPhones, display Classic Bluetooth only in "Settings" -> "Bluetooth" (by default), and the Bluetooth LE advertisement will be filtered out.
- It is recommended to use a dedicated Bluetooth LE application to debug the Bluetooth LE function. For example, LightBlue application can be used on iPhone.
- Please check whether the advertising packet conforms to the specified format. Mobile phones tend to filter out packets that do not conform to the specified format and display only the correct ones.

Can I process OTA through Bluetooth LE on ESP32?

Yes, please refer to the instance BLE_OTA.

What is the throughput of ESP32 Bluetooth® LE?

- The throughput of ESP32 Bluetooth LE depends on various factors such as environmental interference, connection interval, MTU size, and the performance of peer devices.
- The maximum throughput of Bluetooth LE communication between ESP32 boards can reach 700 Kbps, about 90 KB/s. For devices that support 2M phy, the maximum throughput can reach 1.4 Mbps, approximately 170 KB/s.
- For details, please refer to the ble_throughput example in ESP-IDF.

Does ESP32 support Bluetooth® 4.2 DLE (Data Length Extension) ?

Yes. Bluetooth® 4.2 DLE is supported in all versions of ESP-IDF, and can be implemented by calling corresponding APIs directly. Please refer to esp_ble_gap_set_pkt_data_len.

How can I get ESP32 Bluetooth® Compatibility Test Report?

Please contact sales@espressif.com.

What is the transmit power of ESP32 Bluetooth LE?

The transmit power of ESP32 Bluetooth LE has 8 levels, corresponding to the power of -12 ~ 9 dBm, with an interval of 3 dBm per level. The configurable transmit power varies for different chip series, please refer to the corresponding header file of the chip. For example: ESP32.

Can ESP32 realize bridging between Wi-Fi and Bluetooth® LE?

Yes, this function is developed on the application layer. You can retrieve data through BLE Mesh and send them out via Wi-Fi. For detailed information, please refer to ESP BLE Mesh and WiFi Coexist example, and modify it for your own application.

What is the operating current of ESP32 Bluetooth® LE?

Current	MAX (mA)	Min (mA)	Average (mA)
Advertising: Adv Interval = 40 ms	142.1	32	42.67
Scanning: Scan Interval = 160 ms, Window = 20 ms	142.1	32	44.4
Connection(Slave): Connection Interval = 20 ms, latency = 0	142.1	32	42.75
Connection(Slave): Connection Interval = 80 ms, latency = 0	142.1	32	35.33

What kinds of Bluetooth® LE profiles does ESP32 support?

Currently, ESP32 Bluetooth LE fully supports some basic profiles, such as GATT/SMP/GAP, as well as some self-defined profiles. The ones that have already been implemented include Bluetooth LE HID (receiving side), Bluetooth LE SPP-Like, Battery, DIS, BluFi (Bluetooth Network Configuration-transmitting side), and so on.

How does ESP32 Bluetooth® LE enter Light-sleep mode?

On the hardware level, a 32 kHz external crystal should be added, or the Light-sleep mode will not take effect.

On the software level (SDK4.0 and later versions), the following configurations should be enabled in menuconfig:

- Power Management:| `menuconfig > Component config > Power management > [*] Support for power management`
- Tickless Idle:| `menuconfig > Component config > FreeRTOS > [*] Tickless idle support (3) Minimum number of ticks to enter sleep mode for (NEW)`

Note: Tickless idle needs to be enabled to allow automatic light-sleep mode. FreeRTOS will enter Light-sleep mode if no tasks need to run for 3 ticks (by default), that is 30 ms if tick rate is 100 Hz. Configure the FreeRTOS tick rate to be higher if you want to allow shorter duration of light-sleep mode, for example: `menuconfig > ``Component config > FreeRTOS > (1000) Tick rate (Hz).`

- Configure external 32.768 kHz crystal as RTC clock source :| `menuconfig > Component config > ESP32-specific > RTC clock source (External 32 kHz crystal)[*] Additional current for external 32 kHz crystal`

Note: The "additional current" option is a workaround for a hardware issue on ESP32 that the crystal can fail in oscillating. Please enable this option when you use external 32 kHz crystal. This hardware issue will be resolved in the next chip revision.

- Enable Bluetooth modem sleep with external 32.768kHz crystal as low power clock :| `menuconfig > Component config > Bluetooth > Bluetooth controller > MODEM SLEEP Options > [*] Bluetooth modem sleep`

Are there any documentation references for ESP32 BluFi networking?

For BluFi networking, please refer to ESP32 Blufi. For BluFi networking examples, please refer to Blufi.

How many Bluetooth LE clients can be connected to ESP32?

The Bluetooth® LE Server supports up to nine client connections at most, please check the configuration parameter `ble_max_conn` in the application. For stable connection, three clients should be good. If you need to support more connected devices, you can choose ESP32-C6/C5/H2. For the multi-connection capabilities of different chips, you can refer to the Multi-Connection Guide.

How can I get the MAC address of Bluetooth® devices for ESP32?

You can get the MAC address configured by Bluetooth via API esp_bt_dev_get_address(void);, also the system pre-defined MAC address types via API esp_err_t esp_read_mac(uint8_t* mac,esp_mac_type_ttype);.

Is it possible to use Wi-Fi Smartconfig and Bluetooth® LE Mesh for ESP32 simultaneously?

It is not recommended to use them simultaneously.

- The Smartconfig will need to receive the networking data, thus occupying the antenna all the time. If it is used together with Bluetooth LE Mesh, there will be an extremely high rate of failure.
- The Bluetooth LE Mesh can be used together with BluFi. So it is recommended to use BluFi for networking.

How can I modify the transmit power of Bluetooth® LE for ESP32 series chips ?

- For ESP32/ESP32-S3/ESP32-C3, the Bluetooth transmit power can be configured via function `esp_ble_tx_power_set()`. For details, please refer to esp_bt.h.
- For ESP32-C6/ESP32-C2/ESP32-H2, you can set the transmit power by calling the esp_ble_tx_power_set_enhanced() API.

How is the networking compatibility of ESP32 Bluetooth® LE? Is it open-sourced?

- ESP32 Bluetooth networking, BluFi networking for short, has a good compatibility as Bluetooth LE and is compatible with many mainstream mobile phones such as Apple, HUAWEI, Mi, OPPO, MEIZU, OnePlus, ZTE and etc.
- Currently, the BluFi protocol and phone application code is open-sourced.

How is the security of ESP32 Bluetooth® LE/Bluetooth® Secure Simple Pairing (SSP) compared to legacy pairing?

- Secure Simple Pairing (SSP) is more secure than legacy pairing.
- The legacy pairing uses symmetric encryption algorithm, while Secure Simple Pairing (SSP) uses asymmetric cryptography algorithm.

How can I confirm the MTU size of ESP32 Bluetooth® LE?

- By default, the MTU size of ESP32 Bluetooth LE is 23 bytes, and can be configured to reach 517 bytes.
- For phones, the MTU size can be self-defined. Then, the end device with a smaller MTU will be chosen for communication.

Does ESP32 Bluetooth® LE support Client-Server mode, in which gatt server and gatt client can coexist?

- Yes, please refer to example gattc_gatts_coex.

What are the risks if there are over six devices connected to ESP32 Bluetooth® LE?

- Usually it depends on the specific application scenario. In general, the ESP32 Bluetooth LE can communicate stably with three devices connected.
- There is no exact number for maximum Bluetooth LE connections. When there are multiple devices connected to Bluetooth LE simultaneously, the RF is time-multiplexed, thus requiring the designer to ensure that each device is not overly occupied, causing other devices to timeout and disconnected.
- The connection parameters include: connection interval, connection window, latency and timeout. It is ok for devices to not respond within the `latency`, but if the responding time exceeds `timeout` threshold, the device will be disconnected.
- If the `interval` is configured to 100 and `window` to 5, the Bluetooth LE will be able to connect to more devices with Wi-Fi disconnected. However, If Wi-Fi is connected and the value of `interval` is too small, only a few devices can be connected.
- When the Bluetooth LE supports multiple devices connected simultaneously, there will be bigger possibility for RF solt management to generate error. So when there are multiple connections for Bluetooth LE, it is necessary to debug for different scenarios.

When using ESP32 device as the server of Bluetooth® LE, how many client devices can be connected?

- The ESP32 Bluetooth LE supports up to nine client devices for connection. It is recommended to hold this number within three.
- Please make configurations via `menuconfig` > `Component config` > `Bluetooth` > `Bluetooth controller` > `BLE MAX Connections`.

When I use the "BluFi" example to configure network for ESP32, the Wi-Fi cannot be connected during the distribution process via the EspBluFi application since a wrong Wi-Fi has been configured. Then the device is restarted after sending a SCAN command from the application. What is the reason?

- The "BluFi" example stipulates that Wi-Fi "SCAN" commands cannot be sent when Wi-Fi is connected.
- To solve this issue, you can add `ESP_ERROR_CHECK(esp_wifi_disconnect());` to the first line of the `ESP_BLUFI_EVENT_GET_WIFI_LIST:{};` function under the `blufi_example_main.c` file.

When I set name for the bluetooth of an ESP32 device using Chinese characters, messy code shows instead. What is the reason ?

- This is because the Chinese encoding format of the editor is not UTF-8 at this time, and the encoding format of the editor needs to be changed to UTF-8.

When I upload sub-packages to the Bluetooth channel using ESP32, the maximum transmission data length of a packet is 253 (MTU is set to 263). This results in slower transmission when a large number of data packets are transmitted for multi-packet reading. Is there a BluFi extension protocol that can support the transmission of a larger length of data in one packet, or are there other solutions to increase the transmission rate?

- The transmission is slow When a large number of data packets on the Bluetooth channel are transmitted for multi-packet reading. You can improve the transmission speed by adjusting the Bluetooth connection parameters.
- The BLE packet length setting depends on the `ESP_GATT_MAX_MTU_SIZE` setting, please refer to the Description.
- The configured MTU size will affect the data transmission rate. The effective MTU length needs to be changed by MTU exchange to change the default MTU size. The MTU size used in the final MTU exchange is used as the MTU size for the communication between the two devices. You can check the value of the MTU after exchange, such as the follows:

```
case ESP_GATTS_MTU_EVT:
        ESP_LOGI(GATTS_TAG, "ESP_GATTS_MTU_EVT, MTU%d", param->mtu.mtu);
```

How many stable connections can be reached for ESP32-C3' s Bluetooth® LE?

Depending on the connection parameters, no more than eight are allowed. For better multi-connection performance, please choose ESP32-C6 series.

How can I adjust the BLE advertising interval?

- The advertising interval is decided by `adv_int_min` and `adv_int_max` parameters in BLE advertising struct, which configures the minimum and maximum advertising interval respectively.
- The advertising interval ranges from 0x0020 to 0x4000 and the default value is 0x0800. The interval time is the value * 0.625 ms, i.e., 20 ms to 10.24 sec.
- If the values of `adv_int_min` and `adv_int_max` are different, the advertising interval is within the range of the two values. If the values are the same, the interval will be this fixed value.

How much memory does ESP32 Bluetooth occupy?

- Controller:
  - BLE single mode: 40 KB
  - BR/EDR single mode: 65 KB
  - Dual mode: 120 KB
- Main equipment:
  - BLE GATT Client (Gatt Client demo): 24 KB (.bss+.data) + 23 KB (heap) = 47 KB
  - BLE GATT Server (GATT Server demo): 23 KB (.bss+.data) + 23 KB (heap) = 46 KB
  - BLE GATT Client & GATT Server: 24 KB (.bss+.data) + 24 KB (heap) = 48 KB
  - SMP: 5 KB
  - Classic Bluetooth (Classic Bluetooth A2DP_SINK demo, including SMP/SDP/A2DP/AVRCP): 48 KB (.bss+.data) + 24 KB (heap) = 72 KB (an additional 13 KB is added when the example is running)

Note: The above heap (Heap) all include the task stack (Task Stack), because the task stack is allocated from the heap and considered as a heap.

- Optimized PSRAM version:

In ESP-IDF v3.0 and later versions, if you open the PSRAM related options of the Bluetooth menu in `menuconfig`, and put part of the .bss/.data section and heap of Bluedroid (Host) into PSRAM, almost 50 KB memory space can be saved.

When I use the "gattc_gatts_coex.c" example on ESP32 to test BLE multi-connection, it can only connect to four devices even after I set the `BLE Max connections` in `menuconfig` to five. What is the reason?

- Please set the `BT/BLE MAX ACL CONNECTION` in `menuconfig` to five.

Does ESP32-C3 BLE support master and slave mode at the same time? What is the number of connections in master mode and slave mode?

- ESP32-C3 supports master and slave mode at the same time, which share 8 connections. For example, if ESP32-C3 connects to 4 slave devices, it can be connected by 8 - 4 = 4 master devices.
- In addition, when ESP32-C3 is used as a slave, it can be connected by 8 master devices; when used as a master, it can connect to 8 slave devices.

How does BLE capture packets?

- There are many available tools, such as:
  - TI Packet sniffer
  - NRF Packet sniffer

When I use ESP32, can Light-sleep mode be enabled for Bluetooth and can Bluetooth be kept connected in Light-sleep mode?

- To use ESP32 in light-sleep mode, release/4.0 or above versions of ESP-IDF and a 32.768 kHz crystal are needed. For the ESP32 series chips, adding a 32.768 kHz can significantly reduce power consumption.
- Bluetooth can be kept connected in Light-sleep mode. Please refer to Bluetooth modem sleep with external 32.768 kHz xtal under light sleep.

How can I modify the Bluetooth device name of ESP32?

- The structure to be modified is as follows:

```
static uint8_t raw_adv_data[] = {

        /* flags*/

        0x02, 0x01, 0x06,

        Tx power*/

        0x02, 0x0a, 0xeb,

        /* service uuid*/

        0x03, 0x03, 0xFF, 0x00,

        /* device name*/

        0x0f, 0x09,'E','S','P','_','G','A','T','T','S','_','D','E ','M','O'

};
```

- The above `/* device name*/` is the modified item. Among them, 0x0f is the total length of the field type plus specific content, and 0x09 indicates that this type refers to the device name. Subsequent ' E ' ,' S ' ,' P ' ,' _ ' ,' G ' ,' A ' ,' T ' ,' T ' ,' S ' ,' _ ' ,' D ' ,' E ' , ' M ' ,' O ' are the ASCII code of the broadcast device name.
- For more tutorials, please refer to the Device Discovery guide.

What is the maximum supported broadcast length of BLE 5.0 broadcast after it is set to legacy mode?

- The maximum supported length is 31-byte.

How can I set a BLE broadcast package as unconnectable package?

- please refer to the gatt_server demo , and set adv_type as ADV_TYPE_NONCONN_IND.

```
static esp_ble_adv_params_t adv_params = {
  .adv_int_min        = 0x20,
  .adv_int_max        = 0x40,
  .adv_type           = ADV_TYPE_NONCONN_IND,
  .own_addr_type      = BLE_ADDR_TYPE_PUBLIC,
  //.peer_addr            =
  //.peer_addr_type       =
```

(continues on next page)

(continued from previous page)

```
.channel_map        = ADV_CHNL_ALL,
.adv_filter_policy = ADV_FILTER_ALLOW_SCAN_ANY_CON_ANY,
}
```

How many devices can be connected at the most as suggested by the White List of ESP32 Bluetooth LE?

- The maximum supported number is 12.

Can ESP32 Bluetooth LE use PSRAM?

To enable Bluetooth LE to use PSRAM, please go to `Component config > Bluetooth > Bluedroid Options` and enable BT/BLE will first malloc the memory from the PSRAM。

When using ESP32-C3 BLE Scan, can I set it to only scan the Long Range devices?

- Yes, you can make tests based on esp-idf/examples/bluetooth/bluedroid/ble_50/ble50_security_client. By changing the configuration *.cfg_mask = ESP_BLE_GAP_EXT_SCAN_CFG_UNCODE_MASK | ESP_BLE_GAP_EXT_SCAN_CFG_CODE_MASK* in ext_scan_params to *.cfg_mask = ESP_BLE_GAP_EXT_SCAN_CFG_CODE_MASK*, you can scan the broadcast packets whose primary PHY type is LE CODED PHY.

How do I set the ESP32 BLE Scan to the permanent scan without generating a timeout?

- You can realize this by setting "duration" to 0 before using the esp_ble_gap_start_scanning() function to start BLE Scan.

How can I get RSSI of BLE devices through ESP32?

- You can use the esp_ble_gap_read_rssi() function to get RSSI of connected BLE devices.
- If you want to get RSSI of all scanned BLE devices around, please use the ble_scan_result_evt_param structure in the ESP_GAP_BLE_SCAN_RESULT_EVT event to enable the printing of RSSI.

How can I increase the transmission distance of BLE5.0? How can I set BLE5.0 to long-range mode?

- In practice, the transmission distance of BLE5.0 is about 200 m. It is recommended to refer to the actual test distance. ESP32-S3 supports the features of BLE5.0, and supports long-range communication through Coded PHY (125 Kbps and 500 Kbps) and broadcast extension.
- You can realize long-range communication by using 125 Kbps Coded PHY and increasing the transmit power (tx_power). Refer to the following settings:

```
esp_ble_gap_ext_adv_params_t ext_adv_params_coded = {
   .type = ESP_BLE_GAP_SET_EXT_ADV_PROP_SCANNABLE,
   .interval_min = 0x50,
   .interval_max = 0x50,
   .channel_map = ADV_CHNL_ALL,
   .filter_policy = ADV_FILTER_ALLOW_SCAN_ANY_CON_ANY,
   .primary_phy = ESP_BLE_GAP_PHY_CODED,
   .max_skip = 0,
   .secondary_phy = ESP_BLE_GAP_PHY_CODED,
   .sid = 0,
   .scan_req_notif = false,
   .own_addr_type = BLE_ADDR_TYPE_RANDOM,
   .tx_power = 18,
};
```

- For the BLE5.0 examples, please refer to ble_50 examples in ESP-IDF.

I have changed the name of the Bluetooth device with esp_ble_gap_set_device_name() in ESP32-C3. It works for Android devices and the customized device name can be shown. However, it does not work on IOS devices. The device name is still the default Bluetooth name. How can I make it work on Apple devices as well?

- In this case, you need to use raw data to create BLE advertising packets. First, enable the `CONFIG_SET_RAW_ADV_DATA` option in `menuconfig` (`idf.py menuconfig` > `Example 'GATT SERVER' Config` > `Use raw data for advertising packets and scan response data`), and then customize Broadcast packet structure in the gatt server example.
- Please use nRF Connect APP to test. We have tested and it works on the nRF connect APP. This issue is related to IOS APPs.

I want to use two ESP32 development boards to test the Bluetooth connection. How can I set the specified key to automatically connect them with gatt_security_client and gatt_security_server examples?

- In gatt_security_client and gatt_security_server examples, the default key is 123456. For details, please refer to uint32_t passkey = 123456. You can also set other passwords.
- Since the ESP32 device has no display or input keyboard by default, the example sets the IO capability to No output No input. For more details, please refer to Gatt Security Server Example Walkthrough.
- To manually input the key, please set esp_ble_io_cap_t iocap in the gatt_security_server example to ESP_IO_CAP_OUT mode, and then you can use the nRF Connect APP to establish a connection with the BLE Server.

After setting gatt_security_server to ESP_IO_CAP_OUT mode and setting gatt_security_client to ESP_IO_CAP_OUT mode, I deliberately set the wrong passkey. However, the two development boards can still be connected. What is the reason?

- When the server is set to ESP_IO_CAP_OUT mode, gatt_security_client should be set to ESP_IO_CAP_IN mode.
- To avoid such a situation, please add the following code into the case ESP_GAP_BLE_PASSKEY_REQ_EVT event on the gatt_security_client side:

```
esp_ble_passkey_reply(param->ble_security.ble_req.bd_addr, true, 123457);
```

Does ESP32-C3/ESP32-C6/ESP32-S3 support Bluetooth AOA/AOD?

ESP32-C3/ESP32-C6/ESP32-S3 do not support Bluetooth AOA/AOD. For information on the BLE support status of other chips, please refer to Major Feature Support Status.

What is the maximum length of data in a BLE advertising packet supported by ESP32-C3 with the BLE5.0 feature?

- The maximum length is 1650 bytes, which can be set via the esp_ble_gap_config_ext_adv_data_raw() API.

Does ESP32 have any API to check whether BLE advertising has started or stopped?

- For bluedroid stack, there is no such API currently.
- For Nimble stack (and using non-extended advertising of BLE 4.2), you can use the ble_gap_adv_active API.

Does ESP32 support multiple clients connecting at the same time when used as a BLE server? How to realize it?

- ESP32 can be used as a BLE server supporting multiple BLE clients to access simultaneously. Meanwhile, it can also be used as a BLE client connecting to multiple BLE servers simultaneously.
- When used as a BLE server, you can simply enable advertising again after a client connects. Take gatt_server_service_table as an example, after receiving the `ESP_GATTS_CONNECT_EVT` event, please call `esp_ble_gap_start_advertising()` to enable advertising.
- When used as a BLE client, please refer to gattc_multi_connect.

How to set the continuous scanning time for BLE5.0?

- You can use the esp_err_t esp_ble_gap_start_ext_scan(uint32_t duration, uint16_t period); API for configuration. When the period is set to 0, the duration time is the continuous scanning time.

How to set up the GATT service with a 128-bit UUID based on the GATT Server example?

You can refer to the following code:

```
static const uint8_t pctool_service_uuid[16] = {
    0x00, 0x03, 0xcd, 0xd0, 0x00, 0x00, 0x10, 0x00, 0x80, 0x00,␣
↪0x00, 0x80, 0x5f, 0x9b, 0x01, 0x31
};
static const uint8_t pctool_write_uuid[16] = {
    0x00, 0x03, 0xcd, 0xd2, 0x00, 0x00, 0x10, 0x00, 0x80, 0x00,␣
↪0x00, 0x80, 0x5f, 0x9b, 0x01, 0x31
};
/* Full Database Description - Used to add attributes into the␣
↪database */
static const esp_gatts_attr_db_t gatt_db[HRS_IDX_NB] =
{
```

(continues on next page)

(continued from previous page)

```
    // Service Declaration
    [IDX_SVC]        =
    {
{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_16, (uint8_t *)&primary_
↪service_uuid, ESP_GATT_PERM_READ,
      sizeof(pctool_service_uuid), sizeof(pctool_service_uuid),␣
↪(uint8_t *)&pctool_service_uuid}
},
    /* Characteristic Declaration */
    [IDX_CHAR_A]     =
    {
{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_16, (uint8_t *)&character_
↪declaration_uuid, ESP_GATT_PERM_READ,
      CHAR_DECLARATION_SIZE, CHAR_DECLARATION_SIZE, (uint8_t *)&
↪char_prop_read_write_notify}
},
    /* Characteristic Value */
    [IDX_CHAR_VAL_A] =
    {
{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_128, (uint8_t *)&pctool_write_
↪uuid, ESP_GATT_PERM_READ | ESP_GATT_PERM_WRITE,
      GATTS_DEMO_CHAR_VAL_LEN_MAX, sizeof(char_value), (uint8_t␣
↪*)char_value}
},
}
```

When testing based on the GATT Server example, is it possible to delete the default 1800 and 1801 service attributes?

- The 1800 and 1801 service attributes are two standard GATT service attributes in the BLE protocol, and it is not recommended to delete or disable them. They are part of the BLE protocol specifications, providing basic device information and general access capabilities, and maintaining compatibility with the standard BLE protocol.
- 0x1800 refers to generic access, defining the general attributes of the device, while 0x1801 refers to generic attribute, a simple GATT service used to provide basic information about the device.
- If you indeed need to delete these two services, you can comment out the two lines of code gap_attr_db_init() and gatt_profile_db_init().

Is there an explanation for the BLE error codes in the ESP-IDF SDK?

- The BLE error codes in the ESP-IDF SDK refer to the BLE standard protocol. The corresponding error code descriptions can be found in LIST OF BLE ERROR CODES.

The error below occurred when setting the Bluetooth mode to `Component config > Bluetooth > Controller Options > Bluetooth controller mode (BR/EDR/BLE/DUALMODE)` dual mode based on the BLE SPP Server example. What could be the reason for this?

```
E (2906) GATTS_SPP_DEMO: spp_gatt_init enable controller failed:␣
↪ESP_ERR_INVALID_ARG
```

- The current reported error is due to the BLE SPP Server example releases the memory of Class Bluetooth controller by default. Please refer to the API description for esp_bt_controller_mem_release().
- After setting the Bluetooth Dual Mode, you need to delete ESP_ERROR_CHECK(esp_bt_controller_mem_release(ESP_BT_MODE_CLASSIC_BT));, then modify ret = esp_bt_controller_enable(ESP_BT_MODE_BLE); to `ret = esp_bt_controller_enable(ESP_BT_MODE_BTDM);`.

Is there an example of implementing a Bluetooth LE Eddystone beacon based on ESP32?

- Currently, there is no such example. You can implement such an application by modifying the esp-idf/examples/bluetooth/bluedroid/ble/ble_eddystone example and referring to Eddystone Protocol Specification.

Is there an official Bluetooth LE OTA example?

- Yes, see esp-iot-solution/examples/bluetooth/ble_ota.
- In addition, the source code of the Bluetooth LE OTA APP for the Android and IOS versions is now public on GitHub. See Android source code and IOS source code. You need to manually put the bin file to be upgraded into a specific APP path. The README of the corresponding GitHub project provides instructions on where to put the file.

For the ble_ota example in the esp-iot-solution repo, the default Bluetooth protocol stack is Bluedroid. The Android EspBleOTA APP cannot detect the device name `ESP&C919`, while the IOS EspBleOTA APP can. Why?

- For the ble_ota example, the default Bluetooth protocol stack is Bluedroid, and Bluetooth LE 5.0 features are enabled by default. The configuration path is: `menuconfig` > `Component config` > `Bluetooth` > `Bluedroid Options` > `Enable BLE 5.0 features`. However, the current Android EspBleOTA APP does not support scanning Bluetooth LE 5.0 devices, so the device name cannot be scanned by this Android APP.
- Bluetooth 4.0 and 5.0 protocols in the Android system use two APIs to realize the Bluetooth scanning functionality, while the two protocols in the IOS system apply the same API. Thus, the IOS APP can scan the device.
- Some compatible adjustments should be made to scan devices with the Android APP. However, we do not have such development plans currently. If you want to scan the device with the Android APP, please turn off Bluetooth LE 5.0 features.
- Additionally, when the protocol stack is set to Nimble, the Android APP can scan the device name `nimble-ble-ota`. This is because when using Nimble, the Bluetooth LE 5.0 extended broadcast is turned off by default. This can be configured in `menuconfig` > `Example Configuration` > `Enable Extended Adv`.

Does the NimBLE on ESP32-C3 support the Coded PHY mode?

Yes, but only in the ESP-IDF v5.0 and above versions. You can refer to the esp-idf/examples/bluetooth/nimble/ble_phy example, which supports switching between 1 Mbps PHY, 2 Mbps PHY, and Coded PHY (125 Kbps and 500 Kbps).

Does ESP32-S3 support simultaneous broadcastingscanningconnecting under both 125 Kbps Coded PHY and 1 Mbps PHY?

Yes, please refer to the following examples:

- esp-idf/examples/bluetooth/bluedroid/ble_50/multi-adv
- esp-idf/examples/bluetooth/nimble/ble_multi_adv
- esp-idf/examples/bluetooth/nimble/ble_multi_conn/ble_multi_conn_cent

---

How to resolve the error `A stack overflow in stack BTC_TASK has been detected` printed during Bluetooth operation?

This is due to a stack overflow in the Bluetooth controller task. You can increase the stack size in menuconfig: `idf.py menuconfig` -> `Component config` -> `Bluetooth` -> `Bluedriod options` -> `(3072)Bluetooth event(callback to application) task stack size`.

Can ESP-IDF enable both BLE4.2 and BLE5.0 at the same time?

Currently, both BLE4.2 and BLE5.0 can be enabled simultaneously in menuconfig, but the APIs for BLE4.2 and BLE5.0 are different and can not be used interchangeably. Therefore, in the future, it will be modified to only allow either BLE4.2 or BLE5.0 to be enabled, and it will not be possible to enable both at the same time.

When the ESP32 device acts as a BLE server, how does the slave initiate MTU negotiation?

According to the BLE protocol, MTU negotiation must be initiated by the client, and the server cannot initiate it actively. Therefore, the slave can act as a client to initiate MTU negotiation.

How to specify BLE channel?

- For BLE peripheral devices, you can specify the broadcast channel using `channel_map` in `esp_ble_adv_params_t`.
- For BLE central devices, you can call the `esp_gap_ble_set_channels` API to update the channel map. For example, `{0xFF, 0xFF, 0xFF, 0xFF, 0x1F}` means enabling all channels; and `{0xFF, 0xFF, 0xFF, 0xFF, 0x0F}` means blocking channel 36.

ESP32 BLE takes 4 seconds to discover a service UUID. How to accelerate this process?

The time required to discover a service's UUID depends on the BLE connection parameters, the complexity of the GATT service, and the MTU. To shorten the time, you can try to reduce the BLE connection interval, increase the MTU, or narrow the GATT service discovery scope.

How to deinitialize the Nimble application on ESP32-C6?

You can use the following sequence of APIs to deinitialize Nimble:

```
nimble_port_stop();
nimble_port_freertos_deinit();
nimble_port_deinit();
```

How to terminate BLE tasks and release BLE resources on ESP32?

- To terminate a BLE Server task, invoke the following API sequence:

```
esp_ble_gap_stop_advertising ();
esp_ble_gap_disconnect ();
esp_ble_gatts_close  ();
```

- To terminate a BLE Client task, invoke the following API sequence:

```
esp_ble_gap_stop_scanning ();
esp_ble_gap_disconnect ();
esp_ble_gattc_close ();
```

- To release BLE resources, invoke the following API sequence:

```
esp_bluedroid_disable ();
esp_bluedroid_deinit ();
esp_bt_controller_disable ();
esp_bt_controller_deinit ();
```

- For more programming guides, please refer to: Low Power Bluetooth API usage instructions.

How to improve the BLE throughput of the ESP32 series chips?

- The BLE throughput of the ESP32 series products depends on various factors, such as environmental interference, BLE connection interval, MTU size, and the performance of the peer device.
  - The smaller the BLE connection interval, the faster the BLE transmission rate.
  - The larger the MTU size, the faster the BLE transmission rate.
- ESP32-S3 supports BLE 5.0 features, which can be tested based on the ble_throughput example between two ESP32-S3 devices. The reference data is as follows:
  - 1M PHY， 0.73 Mbit/s
  - 2M PHY， 1.35 Mbit/s

When using the official gatt_server_service_table example, the device can be discovered by the phone’ s system Bluetooth but cannot be connected, whereas BLE debugging apps (such as BLE Debug Assistant or nRF Connect) can connect successfully. What is the reason?

- Cause Analysis:
  - The phone’ s system Bluetooth is not a universal BLE client. iOS/Android system Bluetooth is more like **Classic Bluetooth (BR/EDR)** or standard **BLE Profiles** (e.g., HID, HRP, ANCS). System Bluetooth can connect to devices like BLE keyboards, mice, and wearables.
  - System Bluetooth usually initiates a pairing request during connection.
  - The gatt_server_service_table example does not enable BLE encryption and does not implement standard **BLE Profiles**, so the connection cannot be successfully established.
- Solution:

  - To allow the device to be successfully connected by system Bluetooth, BLE encryption must be enabled on the device, and a standard **BLE Profile** should be implemented.
- Reference Examples:
  - gatt_security_server example enables BLE encryption and uses Heart Rate Profile (HRP), allowing the system Bluetooth to connect successfully.
  - ble_hid_device_demo example enables BLE encryption and uses HID Profile, allowing the system Bluetooth to connect successfully.

4.1.2 Classic Bluetooth

Can I process OTA through Classic Bluetooth® on ESP32?

Yes, please operate basing on bt_spp_acceptor and bt_spp_initiator.

How do I connect mobile phones and play music using ESP32 Bluetooth®?

ESP32 is used as an A2DP receiver when connected to a cell phone to play music. Please note that the A2DP Sink Demo uses a mobile phone to obtain SBC encoded data stream only. In order to play sounds, you will also need to decode the data and some peripherals, including codec modules, D/A converter, and speaker.

What is the maximum transmission rate for ESP32 Classic Bluetooth® SPP?

In an open environment, the transmission rate for ESP32 Classic Bluetooth SPP can reach 1400+ Kbps ~ 1590 Kbps (only for reference, please do tests based on your actual application environment) with bi-directional transmitting and receiving simultaneously.

Is ESP32 Bluetooth® compatible to Bluetooth® ver2.1 + EDR protocol?

Yes. The ESP32 Bluetooth is downward-compatible, you can do tests using our official Bluetooth examples.

What is the operating current for ESP32 Classic Bluetooth® ?

A2DP (160 MHz dual-core CPU, DFS disabled, light-sleep mode disabled, commit b7159741)

Current	Maximum (mA)	Minimum (mA)	Average (mA)
Sniff interval = 33.75 ms	183.1	30.27	42.5
Play Music	192.62	–	103.03

A2DP (160 MHz dual-core CPU, DFS enabled, light-sleep mode enabled, commit b7159741)

Current	Maximum (mA)	Minimum (mA)	Average (mA)
Sniff interval = 33.75 ms	196.12	1.47	19.7
Play Music	219.4	–	101.26

Since the CPU is constantly working while playing music, no minimum value has been recorded.

How can I modify the transmit power of Classic Bluetooth® for ESP32 series chips ?

Use esp_bredr_tx_power_set() for setting the transmit power.

When I execute example bt_spp_acceptor on ESP32, the IOS device cannot find the ESP32 device during scanning. What could be the reasons?

- Apple has opened Bluetooth® as: A2DP, HID ' s keyboard, avrcp, SPP (need MFI), high-level Bluetooth LE and ANCS for Bluetooth LE.
- If the IOS device expects to communicate with the end device via SPP, the SPP of the end device should have the MFI certificate. However, ESP32 SPP does not have the MFI certificate, thus the IOS device cannot find ESP32 during scanning.

How can I send files via Bluetooth® BR/EDR for ESP32?

- Please refer to example `bt_spp_acceptor` or `bt_spp_initiator` in classic bt.

What profile does ESP32 ' s classic Bluetooth® support?

- v5.3 and below versions: Support for A2DP, AVRCP, SPP, HFP, HID.
- v5.4 and above versions: Support for A2DP, AVRCP, SPP, HFP, HID, PBAP-Client.

How can I input the PIN code via mobile phone during ESP32 ' s Classic Bluetooth Pairing mode?

You can disable `Secure Simple Pairing` to support only `Legacy Pairing`.

- From esp-idf v3.3 to v4.0 (not include v4.0): `Component config` > `Bluetooth` > `Bluedroid Enable` > `[*] Classic Bluetooth` > `[ ]Secure Simple Pairing`
- v4.0 to v5.2: `Component config` > `Bluetooth` > `Bluedroid Options` > `[ ] Secure Simple Pairing`
- v5.3 and above versions: You need to call the `esp_bluedroid_init_with_cfg()` function for configuration. To be specific: create a configuration structure of type `esp_bluedroid_config_t`, and set its `ssp_en` member to false, then pass this structure to the initialization function.

What is the maximum MTU Size of ESP32 Classic Bluetooth?

- ESP32 Classic Bluetooth has two protocols, namely A2DP and SPP. The maximum MTU Size setting of BT A2DP (default) is 1008 bytes, of which the header occupies 12 bytes and the actual amount of data transmitted by the application layer is 1008 - 12 = 996 (bytes); the maximum MTU Size of BT SPP (default) Set to 990 bytes.

Does ESP32 support transmitting audio stream using A2DP?

Yes, please refer to example a2dp_source.

Does ESP32 Classic Bluetooth support AVRCP 1.5 or AVRCP 1.6?

AVRCP 1.5 is currently supported on esp-idf v5.0.4 to v5.3, and AVRCP 1.6 is supported on v5.4 and later versions. For details, please refer to esp-idf/components/bt/host/bluedroid/stack/avrc/avrc_sdp.c.

Does ESP32 Classic Bluetooth support AAC?

Currently, this function is not supported.

Does ESP32 support BT mesh functionality?

Not supported. Classic Bluetooth is not suitable for mesh functionality as it primarily supports point-to-point communication.

Does ESP32 classic Bluetooth support the Bluetooth printing service (HCRP) on Windows?

This function is currently not supported.

4.1.3 Other Bluetooth

How can I send Bluetooth HCI commands directly to ESP32-WROOM-32D module through the serial port?

- Please refer to controller_hci_uart_esp32.
- When ESP32 is used as a controller, and the other device serves as a host, HCI commands can be sent to ESP32 via UART.

4.2 Ethernet

4.2.1 When building an example on ESP32 Ethernet development board, an error occurred as "emac: Reset EMAC Timeout" . What could be the reasons?

This is because the EMAC initialization is timeout, and is possibly related to the RMII clock. It is recommended to check your hardware, e.g., see if the PHY crystal oscillator is a cold joint.

4.2.2 When ESP32 connected to LAN8720 externally, with GPIO0 providing CLK, the initialization of Ethernet example failed. How to resolve such issue?

```
I (229) cpu_start: App cpu up.
I (247) heap_init: Initializing. RAM available for dynamic allocation:
I (254) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (260) heap_init: At 3FFB40A8 len 0002BF58 (175 KiB): DRAM
I (266) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (273) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (279) heap_init: At 400885D0 len 00017A30 (94 KiB): IRAM
I (285) cpu_start: Pro cpu start user code
I (303) cpu_start: Chip Revision: 1
W (303) cpu_start: Chip revision is higher than the one configured in␣
↪menuconfig. Suggest to upgrade it.
I (307) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
I (319) system_api: Base MAC address is not set, read default base MAC␣
↪address from BLK0 of EFUSE
E (1329) emac: Timed out waiting for PHY register 0x2 to have value␣
↪0x0007(mask 0xffff). Current value 0xffff
E (2329) emac: Timed out waiting for PHY register 0x3 to have value␣
↪0xc0f0(mask 0xfff0). Current value 0xffff
E (2329) emac: Initialise PHY device Timeout
ESP_ERROR_CHECK failed: esp_err_t 0xffffffff (ESP_FAIL) at 0x40084140
0x40084140: _esp_error_check_failed at /mnt/hgfs/workspace/esp32/IDF/esp-
↪idf-v3.3/components/esp32/panic.c:720

file: "/mnt/hgfs/workspace/esp32/project/ethernet/main/ethernet_example_
↪main.c" line 153
func: app_main
expression: esp_eth_enable()

ELF file SHA256:␣
↪``597d55ebf237c1cffa5f47c73148a159b22726d94a7b78100bd941d7d5fc906e``

Backtrace: 0x40083cdc:0x3ffb5e80 0x40084143:0x3ffb5ea0␣
↪0x400d32c1:0x3ffb5ec0 0x400d1742:0x3ffb5f20 0x40085d91:0x3ffb5f40
0x40083cdc: invoke_abort at /mnt/hgfs/workspace/esp32/IDF/esp-idf-v3.3/
↪components/esp32/panic.c:715

0x40084143: _esp_error_check_failed at /mnt/hgfs/workspace/esp32/IDF/esp-
↪idf-v3.3/components/esp32/panic.c:721

0x400d32c1: app_main at /mnt/hgfs/workspace/esp32/project/ethernet/main/
↪ethernet_example_main.c:153 (discriminator 1)
```

(continues on next page)

(continued from previous page)

```
0x400d1742: main_task at /mnt/hgfs/workspace/esp32/IDF/esp-idf-v3.3/
↪components/esp32/cpu_start.c:542

0x40085d91: vPortTaskWrapper at /mnt/hgfs/workspace/esp32/IDF/esp-idf-v3.
↪3/components/freertos/port.c:403


Rebooting...
```

- Please check if there are capacitors on IO0. It is better to have no capacitors on IO0 when it is used as the CLK output pin, or it would affect the timing.
- When GPIO0 starts to output RMII clock, please remember to check the `CONFIG_PHY_CLOCK_GPIO0_OUT` option in Kconfig.
- For references about Ethernet, besides the README in example, you can also refer to API Reference.

4.2.3 When using the Ethernet example in ESP-IDF, an error occurred as "Timed out waiting for PHY register 0x3 to have value 0xc0f0 (mask 0xfff0). Current value 0xffff" . How to resolve such issue?

- Please refer to BBS issue and Github issue.
- When the value of the PHY register is 0xFFFF, please check the following:
  a. If the wiring of MDIO and MDC is correct
  b. If the 50 MHz clock required by RMII is normal
  c. If the address of PHY (both software and hardware) is correctly configured
- It is strongly recommended to check the strapping pins that control the PHY address to make sure they are not floating and **not in the default state**. Please make sure they have already been pulled up or down by external resistance.
- If you are still not sure about the PHY address, you can try to set the PHY address from 0 to 31, and read the PHY ID register to see if you can get the correct data. If yes, then write down the current PHY address.

4.2.4 When using ESP-IDF v4.1, how to set the static IP for ESP32 Ethernet?

Since v4.1 and later versions of ESP-IDF will remove the tcp/ip interface, it is recommended to use the ESP-NETIF interface.

Code example:

```
{
    ...
    esp_netif_config_t cfg = ESP_NETIF_DEFAULT_ETH();
    esp_netif_t *eth_netif = esp_netif_new(&cfg);
    // Set default handlers to process TCP/IP stuffs
    ESP_ERROR_CHECK(esp_eth_set_default_handlers(eth_netif));
    ...
    char* ip= "192.168.5.241";
    char* gateway = "192.168.5.1";
    char* netmask = "255.255.255.0";
    esp_netif_ip_info_t info_t;
    memset(&info_t, 0, sizeof(esp_netif_ip_info_t));

    if (eth_netif)
```

(continues on next page)

(continued from previous page)

```
    {
        ESP_ERROR_CHECK(esp_netif_dhcpc_stop(eth_netif));
        info_t.ip.addr = esp_ip4addr_aton((const char *)ip);
        info_t.netmask.addr = esp_ip4addr_aton((const char *)netmask);
        info_t.gw.addr = esp_ip4addr_aton((const char *)gateway);
        esp_netif_set_ip_info(eth_netif, &info_t);
    }
    ...
}
```

4.2.5 Is there any impact on Ethernet functionality if replacing the module of ESP32-Ethernet-Kit with ESP32-WROOM-32D?

- The ESP32-WROVER-B of ESP32-Ethernet-Kit can be replaced with ESP32-WROOM-32D, and its Ethernet functionality will not be affected.
- The main difference between ESP32-WROOM and ESP32-WROVER module series: ESP32-WROVER modules have a 4 MB PSRAM while ESP32-WROOM modules do not have any PSRAM by default. Please refer to:
  - ESP32-WROOM-32D Datasheet
  - ESP32-WROVER-B Datasheet
- The ESP32-WROOM and ESP32-WROVER modules all use the ESP32 chip as their core, which supports Ethernet. For more information, please refer to ESP32 Datasheet.
- Related document: ESP32-Ethernet-Kit Getting Started Guide.

4.2.6 When using ESP32 to design a self-developed Ethernet board, after downloaded the official esp-idf/examples/ethernet example, errors are reported as follows, what is the reason?

```
E (5556) emac: Timed out waiting for PHY rdgister 0x2 to have value␣
↪0x0022 (mask 0xffff). Current value 0xffff
E (6556) emac: Timed out waiting for PHY register 0x3 to have value␣
↪0x1430 (mask 0xfff0). Current value 0xffff
```

- This error indicates something is wrong with your hardware circuit. The RMII clock is not working normally with the PHY, causing the PHY failed to read registers. For the more information about RMII clock, please refer to Instructions.

4.2.7 Does ESP32 Ethernet support MII interface?

- Supported on hardware level, software adaptation is in development. Please refer to Ethernet doc for self implementation.

4.2.8 Is it possible to connect ESP32-S2 to Ethernet externally?

- Yes, ESP-IDF currently provides drivers for the DM9051 module, which has integrated Ethernet MAC and PHY functionality and can communicate with the MCU via the SPI interface. The DM9051 has an integrated MAC+PHY module, please refer to example reference <https://github.com/espressif/esp-idf/tree/master/examples/ethernet/> `_ and `API reference.

4.2.9 Do the ESP32 series chips support to use EMAC and SPI-Ethernet modules simultaneously?

- Yes. ESP32 series chips support to use EMAC and one or two SPI-Ethernet modules simultaneously. You can start PHY and SPI-Ethernet modules simultaneously in menuconfig to test by referring to the example esp-idf/examples/ethernet/basic.

4.2.10 How to diagnose packet delay issues in ESP32 Fast Ethernet transmission?

Please add debug prints in `ethernetif.c` file in the ESP-IDF repository, focusing on checking the TCP sequence number and acknowledgment number (ACK), to help determine whether the delay comes from the Ethernet layer or the LWIP layer.

4.3 coexistence

4.3.1 When Wi-Fi coexists with Bluetooth, what mode does it support?

For supported coexistence scenarios, please refer to coexistence documentation.

4.3.2 When Wi-Fi coexists with ESP-BLE-MESH, the Wi-Fi throughput is low, why?

For ESP32-DevKitC boards without PSRAM, Wi-Fi can coexist with ESP-BLE-MESH but with a relatively low throughput. For ESP32-DevKitC boards with PSRAM, the transmit rate can stabilize at over 1 Mbps.

To support PSRAM, the following configurations in menuconfig should be enabled accordingly:

- `Component config -> ESP PSRAM -> Support for external, SPI-connected RAM -> SPI RAM config -> Try to allocate memories of WiFi and LWIP in SPIRAM firstly. If failed, allocate internal memory`
- `Component config -> Bluetooth -> Bluedroid Options -> BT/BLE will first malloc the memory from the PSRAM`
- `Component config -> Bluetooth -> Bluedroid Options -> Use dynamic memory allocation in BT/BLE stack`
- `Component config -> Bluetooth -> Controller Options -> BLE full scan feature supported`
- `Component config -> Wireless Coexistence -> Software controls WiFi/Bluetooth coexistence (Enable)`

4.3.3 Does ESP32 support coexistence between ESP-WIFI-MESH and Bluetooth® LE Mesh?

No.

However, the ESP32 supports coexistence between ESP-WIFI-MESH and Bluetooth LE, or Wi-Fi STA and Bluetooth LE Mesh.

4.3.4 Does ESP32 support coexistence between Bluetooth® and Wi-Fi?

Yes, but time-sharing control is required for ESP32 ' s coexistence between Wi-Fi and Bluetooth. Please go to menuconfig to enable the Wi-Fi/Bluetooth coexistence, shown as follows:

IDF: release/v5.0

```
menuconfig -> Component config -> Wi-Fi -> Software controls WiFi/
Bluetooth coexistence (Enable)
```

IDF: release/v5.1 and later versions

```
menuconfig -> Component config -> Wireless Coexistence -> Software
controls WiFi/Bluetooth coexistence (Enable)
```

4.3.5 When Wi-Fi, Bluetooth® LE, and A2DP sink coexist, audio data reception is lost and lagged while entering Bluetooth LE scanning. How to resolve such issue?

- Use RingBuf to cache audio data
- Pause music and add a tone, such as "scanning for devices" .

4.3.6 Can BLE advertising (Connectable) and iBeacon sending (advertising) coexist?

IDF: release/v4.0 and later versions | CHIP ESP32:

- Not supported yet on hardware level, but can be realized on application layer by polling and sending broadcast packets at regular intervals.

IDF: release/v4.3 and later versions | CHIP: ESP32-C3|ESP32-S3

- Yes.

4.3.7 How do ESP32 Bluetooth® and Bluetooth® LE dual-mode coexist and how can I use this coexistence mode?

The ESP32 Bluetooth and Bluetooth LE dual-mode does not require complex configurations. For developers, it is simple as calling Bluetooth LE API for Bluetooth LE, and calling Classic Bluetooth API for Classic Bluetooth.

For Classic Bluetooth and Bluetooth LE coexistence, please refer to a2dp_gatts_coex example.

4.3.8 How do ESP32 Bluetooth® and Wi-Fi coexist?

In the `menuconfig`, there is a special option called `Software controls WiFi/Bluetooth coexistence`, which is used to control the coexistence of Bluetooth and Wi-Fi for ESP32 using software, thus balancing the coexistence requirement for controlling the RF module by both the Wi-Fi and Bluetooth modules.

- It is recommended to enable this option if both Bluetooth and Wi-Fi functions are used simultaneously.
- In coexistence scenarios, Bluetooth and Wi-Fi share RF resources through Time Division Multiplexing (TDM). When performing a Bluetooth® LE scan, the scheduled scan window may fall into the Wi-Fi time slot and be interrupted, resulting in an actual scan duration shorter than the configured value. Therefore, it is recommended to set the scan `interval` and `window` to the same value to improve scanning performance.

4.3.9 How can I resolve the frequently occurred ELxXX error (such as ELx200) when Wi-Fi and Ble co-exit ?

CHIP: ESP32

- It has been fixed in commit 386a8e37f19fecc9ef62e72441e6e1272fa985b9. Please switch to the corresponding commit to test.

4.4 Peripherals

4.4.1 Analog-to-Digital Converter (ADC)

What is the resolution of ESP8266 ADC?

- The 10-bit ESP8266 ADC has a theoretical resolution of $2^{10} = 1024$.
- After connected to a router, the ESP8266 will enter Modem-sleep mode from STA mode, causing the change of the reference value inside the chip. Therefore, the ADC could measure the data change.
- If you expect an accurate result, please read the ADC value using function `system_adc_fast_read` after turning off Wi-Fi.

How to get the Bitmap information of the ADC register?

Since the ADC of ESP8266 is highly integrated with the internal RF circuit, the Bitmap and register information is not opened. Please contact sales@espressif.com if you have any special needs.

How many channels does ESP32 ADC have? What is the sampling rate and significant digit?

- The ESP32 ADC has 18 channels.
- If you stop Wi-Fi and use ADC DMA, the sampling rate does not exceed 2 MHz theoretically. However, we recommend you to use a smaller sampling rate in practice.
- Its sampling rate can reach 1000 times per second with Wi-Fi.
- The internal significant digit of ADC is 12 bits.

When calling the API `adc_read_fast()` with ESP8266, will it cause a Wi-Fi disconnection?

- Please turn off Wi-Fi and interrupts first before calling `adc_read_fast()`. Please refer to the Specification of this API.
- Since the `API adc_read_fast()` performs continuous acquisition and the ADC is partially coupled internally with Wi-Fi RF, so it is not possible to call this function with Wi-Fi turned on.
- Please use `adc_read()` for ADC acquisition when Wi-Fi is on. To ensure data stability, you need to use function `esp_wifi_set_ps(WIFI_PS_NONE)` to turn off Wi-Fi Modem-sleep mode.

Note: ADC sampling rate: can reach 100000 times per second with Wi-Fi turned off, and 1000 times per second with Wi-Fi turned on.

If I float the ADC pin and print out VDD3P3 value (65535), then the voltage of VDD3P3 should be 65535/1024 ≈ 63 V. Why this is not the correct voltage value?

The input of ADC should be in the range of 0 V to 3.3 V (the upper limit varies in different chips). The floating measurement is an undefined state.

What is the input resistance of ESP32 ADC?

ADC is capacitive and can be considered as a large resistance.

When using ESP32's ADC to detect the power supply voltage, is it necessary to divide the voltage?

The ADC reference voltage of ESP32 is 1100 mV, but the ADC measurable range can be increased by configuring the internal attenuation. For more information on the measurable range, please refer to ADC Section in the chip datasheet. If the measurable range cannot satisfy your requirement, please add an external voltage division circuit.

What is ESP32's highest sampling rate in ADC DMA mode?

ESP32 supports up to 2 MHz of sampling rate theoretically.

When an ESP32 calling `adc2_get_raw()` between `esp_wifi_start()` and `esp_wifi_stop()`, the read operation fails. What is the reason?

Since Wi-Fi also uses ADC2, and the Wi-Fi driver has a higher priority, the application may fail to read using `adc2_get_raw()` during the operation period of Wi-Fi. It is recommended to check the return value of this function and re-measure it after failure.

Does ESP32 support using ADC2 and Bluetooth simultaneously?

Yes.

What is the sampling rate range supported by the ADC DMA mode of the ESP32-S2 chip?

Frequency limit : 611 Hz ~ 83333 Hz.

Does the ADC of ESP32 support simultaneous sampling of multiple channels?

No, If you are using ADC to do multi-channel sampling, please implement it via ADC polling scheme.

When using the ESP32-WROVER-B module with release/v4.2 version of ESP-IDF, I set the GPIO as an ADC interface, and then set GPIO to other IO mode while with IO mode not effective without any hardware reset, this GPIO does not respond. How do I release the corresponding GPIO mode?

- Please do not set the ADC interface as input-only GPIO.
- When disabling the ADC interface mode, please use adc_digi_stop() to disable the ADC.

What is the measurement error between the ADCs of the ESP32 chip?

By default, the measurement error between ESP32 ADCs is ±6%, please refer to ESP32 datasheet for details.

Can ESP32 measure different data from two ADC channels at the same time, such as current and voltage?

It is not possible to read multiple ADC channels at the same time using one ADC, but you can poll the data of both ADC channels in turn.

Why can't the measured voltage reach the nominal 3100 mV when the ESP32-S3 ADC is configured as `ADC_ATTEN_DB_12`?

When ESP32-S3 ADC1 or ADC2 is configured as `ADC_ATTEN_DB_12`, the voltage measurement range is `0 ~ 3100 mV`. However, the maximum voltage measurement value of some chips is less than `3100 mV`. The following two methods can be used to solve this problem:

- Solution 1: Try to avoid using the boundary voltage values. You can use a divider circuit to reduce the input voltage to an intermediate value for higher accuracy and consistency.
- Solution 2: Use the software ADC Range Extension Solution to extend the maximum measurement voltage to `3300 mV`. This solution is supported in IDF v4.4.8 and v5.3.1 for ESP32-S2 and ESP32-S3 chips, and can be ported to other IDF versions based on this solution.

Can we use GPIO0 as the ADC pin when using ESP32 as a Wi-Fi access point?

- The ESP32 ADC2 pins cannot be used when you are using Wi-Fi. So, if you are having trouble getting the value from an ADC2 GPIO while using Wi-Fi, you may consider using an ADC1 GPIO instead. For more details, please refer to Hardware Limitations of ADC Continuous Mode and Hardware Limitations of ADC Oneshot Mode.
- The GPIO0, GPIO2, GPIO5, GPIO12 (MTDI), and GPIO15 (MTDO) are strapping pins. When using GPIO0 for other functions, you need to pay attention to the GPIO level during power-up. If the GPIO0 level is low during power-up, the chip can enter the download mode. For more information, please refer to ESP32 datasheet.

I tried to test the functionality of ADC2 using GPIO19 and GPIO20 of ESP32-S3 based on "esp-idf/examples/peripherals/adc/oneshot_read" example and set the attenuation parameter of ADC2 to 11 dB. When the input voltage is 0.6 V, why are the test results 1.1 V and 2.8 V?

- Please check whether both two ADC2 channels have been configured according to adc_oneshot_config_channel().

Can ESP32 support some ADC channels in DMA mode and the other ADC channels in oneshot mode under the same ADC controller?

- To use the DMA mode for ESP32 ADC, ESP-IDF v5.0 or later versions should be used.
- ADC2 of ESP32 does not support DMA mode.
- In the same ADC controller, it does not support that some ADC channels are in oneshot mode and other ADC channels are in DMA mode. Please refer to the "ESP32 ADC hardware-limitations".
- In the software, it is recommended to use the adc_continuous_config_t API to set ADC1 as DMA mode and use the adc_oneshot_config_channel API to set ADC2 as oneshot mode.

When using ESP-IDF v5.1 to test ADC2 based on the ESP32-S3-WROOM-1 module, inputting 3.3 V via GPIO12 yet getting a read voltage of 5 V. What could be the reason?

```
I (455346) EXAMPLE: ADC2 Channel[1] Raw Data: 4095
I (455346) EXAMPLE: ADC2 Channel[1] Cali Voltage: 4985 mV
I (456346) EXAMPLE: ADC2 Channel[1] Raw Data: 4095
I (456346) EXAMPLE: ADC2 Channel[1] Cali Voltage: 4985 mV
```

- With normal ADC raw data readings, the reason why ADC conversion value becomes 5 V is because that the effective measurement range of ESP32-S3 ADC is 2900 mV. Please refer to the ESP32-S3 ADC attenuation level corresponding effective measurement range.
- An input voltage exceeding 2900 mV is undefined, which would lead to this situation. If you need to measure an input voltage greater than 2900 mV, it is recommended to use voltage division or adopt the ESP32-S3 ADC Range Extension Solution.

Can the reference source of the ADC be externally applied? If it can be applied, which pin should it be applied to? What is the internal reference voltage of the ADC?

- The ADC does not support external reference voltage. It can only use internal reference voltage.
- The internal reference voltage is 1.1 V.

What is the sample-and-hold capacitance inside the ESP32 SAR-ADC?

The internal sample-and-hold capacitance of the SAR-ADC in the ESP32 chip is approximately 1 pF.

How to use the two-point calibration scheme for ADC on ESP32?

ESP32 uses a reference voltage for ADC calibration by default. To improve the consistency of ADC measurements, you can switch the calibration scheme in eFuse to two-point calibration scheme. However, eFuse changes are irreversible, so please proceed with caution. It is also recommended to use the software-based two-point calibration scheme for ADC: adc_tp_calibration.

4.4.2 Digital-to-Analog Converter (DAC)

When using DAC output for ESP32-S2-Saola-1, the power supply is 3.3 V. But the actual tested voltage is only 3.1 V. Why?

Due to the presence of internal voltage drop and inter-chip differences, even when powered by 3.3 V, the actual maximum output is only around 3.2 V.

4.4.3 GPIO & RTC GPIO

What should I pay attention to for ESP32 pin configurations?

The ESP32 has ESP32-WROOM and ESP32-WROVER series modules. Please pay attention to the following configurations with GPIOs.

The WROOM-32*/32D/32U/32E/32UE series has 26 pins available for customer use, with the following considerations:

- GPIO6-GPIO11 of the WROOM-32*/32D/32U series are occupied by the internal flash and cannot be used for other purposes;
- GPIO6-GPIO11 of the WROOM-32E/32UE series are occupied by the internal flash and are no longer pulled out to the module pins;
- GPIO34, 35, 36 and 39 are input-only pins and cannot be used for outputs;
- ESP32 has a built-in GPIO matrix, and some peripheral interfaces can be connected to any free pin. That is, during hardware design, there is no need to strictly fix certain functions on certain pins;
- WROOM-32*/32D/32U are not recommended for new designs. It is suggested to use the WROOM-32E/32UE series instead.
- In the WROOM-32E/32UE series with QSPI PSRAM, GPIO16 is used to connect to the embedded PSRAM and cannot be used for other functions.

Detailed information can be found in Table 6-2 GPIO_Matrix of ESP32 Series Datasheet.

The WROVER*/WROVER-I*/WROVER-B/WROVER-IB/WROVER-E/WROVER-IE all have 24 pins available for customer use, with the following considerations:

- GPIO6-GPIO11 of the WROVER*/WROVER-I*/WROVER-B/WROVER-IB series are occupied by the built-in flash and cannot be used for other purposes;
- GPIO6-GPIO11 of the WROVER-E/WROVER-IE series are occupied by the built-in flash and are no longer pulled out to the module pins;
- GPIO34, 35, 36 and 39 are input-only pins and cannot be used for outputs;
- For WROVER series, it is not recommended to use GPIO12 for Touch Sensor functions since it has been pulled up in the module;
- ESP32 has a built-in GPIO matrix, and some peripheral interfaces can be connected to any free pin. That is, during hardware design, there is no need to strictly fix certain functions on certain pins;
- WROVER*/WROVER-I*/WROVER-B/WROVER-IB are not recommended for new designs. It is suggested to use the WROVER-E/WROVER-IE series instead.

Detailed information can be found in Table 6-2 GPIO_Matrix of ESP32 Series Datasheet.

There are three sets of UARTs in ESP32, but only UART0 can be used for downloading with fixed pins.

* indicates that the product is in EOL status.

Some ESP8266 GPIOs are high level. What could be the reasons?

- According to the hardware design, some GPIOs are pulled up or down by default. Thus the level of these pins are not controlled by the program during system initialization, causing some incorrect levels of GPIOs during the boot process.
- If you expect to use these GPIOs, it is recommended to keep the hardware peripherals be consistent with the default level status, or adjust level status in software during bootloader process. When using the later method, you may also encounter temporary level exception.

Can I disable the thread scheduling and use a single CPU for ESP32 to realize real-time control of GPIO?

- For now, we do not have any related configurations for SDK to support the single operation of CPU1. Both cores of ESP32 support SMP only, but not AMP.
- The following solutions can be used to resolve the issue of output waveform being interrupted:
  - Use hardware signal outputs, and choose related digital protocols to realize SPI, I2C, I2S and etc. For special usage with SPI, you can generate waveform using signal output lines.
  - See if the hardware RMT can generate the desired waveform with enough length.
  - When the hardware interrupt generated corresponding waveform, all callbacks need to be put in IRAM.
  - Use the co-processor in the chip as a single chip without an operation system.

What is the turning speed of ESP32 GPIO levels?

It takes around 300 ns.

When certain RTC peripherals (SARADC1, SARADC2, AMP, HALL) are powered on, the inputs of GPIO36 and GPIO39 will be pulled down for approximately 80 ns. How to solve the issue?

For applications that require accurate timing and detecting digital input status, the above problems can be avoided by s

- Ignore the inputs from GPIO36 and GPIO39 when turning on the power domain of the above sensors.
- Debounce digital inputs through software. When reading the input states of GPIO36 and GPIO39, debouncing can be implemented through software by sampling and filtering the inputs for multiple times, thus reducing misjudgments caused by short voltage drops.

The ESP32 GPIO peripheral may not trigger interrupts correctly if multiple GPIO pads are configured with edge-triggered interrupts. How to resolve such issue?

- Please search for this question and its answer in ESP32 Series SoC Errata.

Using ESP-WROOM-02D module, can GPIO0, GPIO15, GPIO1 and GPIO3 be used as normal GPIOs?

- Strapping pins (GPIO0 and GPIO15) and download pins (GPIO1 and GPIO3) can be used as normal GPIOs.
- When using the strapping pin as a normal GPIO, you need to pay attention to the level of the strapping pin in the Flash download mode.

After configuring the GPIO19 for ESP32-C3 as pulled-down input, the level of this pin still stays high. However other pins in ESP32-C3 does not have this issue. What is the reason?

- In ESP32-C3, GPIO19 is a USB D+ pin, whose pull-up resistor is controlled by the pin's pull-up value together with USB's pull-up value. If any of the two pull-up values is 1, the pin's pull-up resistor will be enabled.
- The USB pull-up value of GPIO19 is 1 by default, so when the pin is pulled down, GPIO19 still keeps high level.
- This issue has been fixed in the GPIO driver in ESP-IDF v4.4.3 and later versions. For other versions, please write the register `USB_SERIAL_JTAG_DP_PULLUP` to 0 for configuration.

When using the release/v4.2 version of ESP-IDF, how to set a single GPIO as input/output mode simultaneously for ESP32?

You can set via the esp_err_t gpio_set_direction(gpio_num_t gpio_num, gpio_mode_t mode) API.

Is it possible to set the drive capability of the GPIO in ESP-IDF?

Yes. Please use API gpio_set_drive_capability to set the GPIO drive capability.

When ESP32 uses gpio_install_isr_service() to attach a new interrupt service routine on GPIO, why does it return *ESP_ERR_NOT_FOUND*?

Generally, this error means that ESP32 does not have enough available interrupt sources. In this case, there are multiple peripherals occupying the interrupt sources at the same time. You can try to reduce the interrupt sources used by other components to attach new GPIO interrupts.

How do I get the input level of the ESP32 RTC_GPIO?

- You can obtain the input level of RTC_GPIO by reading the macro of the register address corresponding to RTC GPIO. Please refer to "esp-idfcomponentssocesp32includesocrtc_io_reg.h" .
- The related code is as follows :

```
uint8_t level = (uint8_t)((REG_GET_FIELD(RTC_GPIO_IN_REG, RTC_GPIO_IN_
↪NEXT) & BIT(gpio_num)) ? 1 : 0);
```

How to use GPIO buttons in Light-sleep mode?

The wake-up function of the button can be enabled. Please note that non-RTC GPIO cannot enable GPIO edge triggering and level triggering at the same time. You can use the existing functions of the Button component to implement this.

The default state of GPIO20 on ESP32-C6 is initially set as SDIO. How can it be changed to the regular GPIO mode?

You can refer to the following code to change the initial SDIO state of ESP32-C6' s GPIO20 to the regular GPIO mode:

```
gpio_hal_iomux_func_sel(GPIO_PIN_MUX_REG[20], PIN_FUNC_GPIO);
```

Is it possible to set different levels for different GPIOs of ESP32-P4? For example, setting GPIOs in the HP power domain to 3.3 V while setting GPIOs in the LP power domain to 1.8 V?

Mixed settings are not allowed. The levels of all GPIO pins must be uniform, either all at 1.8 V or all at 3.3 V. For example, if a 1.8 V voltage is input to VDDPST_4, then all GPIO levels under this power domain are 1.8 V.

Can the interrupt trigger level in GPIO be adjusted?

Mixed configuration is supported. For example, GPIOs in the VDD_IO_4 domain can use 1.8 V, while GPIOs in the VDD_IO_6 domain can use 3.3 V. However, the voltage levels of the GPIO pins in the same power domain must be uniform, either all 1.8 V or all 3.3 V. For instance, if 1.8 V voltage is input to VDD_IO_4, then all GPIO levels in this power domain are 1.8 V.

How to configure ESP32-C3's GPIO11 (VDD_SPI) pin as a general purpose GPIO in both software and hardware?

- **Hardware:** VDD_SPI is by default used as the power supply pin for the flash. It can only be used as a general-purpose GPIO when the flash is powered by an external power source.
- In software, you can call the esp_efuse_write_field_bit() API to write the eFuse bit of *ESP_EFUSE_VDD_SPI_AS_GPIO* as 1. Refer to the following code:

```
#include "esp_efuse.h"
#include "esp_efuse_table.h"

esp_efuse_write_field_bit(ESP_EFUSE_VDD_SPI_AS_GPIO);
```

When using ESP32-C6 GPIO27 (VDD_SPI) pin as a regular GPIO, what hardware and software operations are required?

- In hardware, VDD_SPI is by default the power supply pin for flash. It can only be used as a regular GPIO when the flash is powered by an external power source.
- Unlike ESP32-C3, ESP32-C6 has some usage restrictions: by default, upon power-up, VDD_SPI will be pulled high. It must be configured through software before it can be used normally. Please refer to the following code:

```
#include "soc/pmu_reg.h"

REG_SET_FIELD(PMU_POWER_VDD_SPI_CNTL_REG, PMU_VDD_SPI_PWR_SEL_SW, 1);
REG_SET_FIELD(PMU_POWER_VDD_SPI_CNTL_REG, PMU_VDD_SPI_PWR_SW, 0);
```

- ESP32-C5 and ESP32-C61 have similar usage restrictions.

4.4.4 Inter-Integrated Circuit (I2C)

Does ESP8266 support I2C slave mode?

Not supported. If you want to use this feature, we recommend using the ESP32 or ESP32-S2 chip. ESP32 example: i2C_self_test.

Is ESP8266 I2C realized via software programming?

Yes, ESP8266 I2C is realized via GPIO software programming.

Is it normal for some spikes to frequently appear on the data line when the ESP32 chip I2C is working (especially in fast mode), often after the falling edge of the 8th/9th clock?

The spike on the data line at the 8th/9th clocks is caused by the I2C master-slave control handover. It is a normal phenomenon and is mentioned in the I2C protocol.

How can the ESP32 chip, as an I2C master, wait for the slave to process the data before receiving it? For example, when reading data through `i2c_master_read_to_device`, the slave needs to return data immediately after accepting the command, but in reality, some slave devices need to wait for a while after receiving the command before they can return data.

This can be realized by dividing `i2c_master_read_device` into the following three steps:

1. Input commands and address: `i2c_cmd_link_create_static` > `i2c_master_start` > `i2c_master_write_byte` > `i2c_master_cmd_begin` > `i2c_cmd_link_delete_static`
2. Delay
3. Read data of the slave: `i2c_cmd_link_create_static` > `i2c_master_read` > `i2c_master_stop` > `i2c_master_cmd_begin` > `i2c_cmd_link_delete_static`

When using the ESP32 chip, can GPIO32 and GPIO33 be configured as I2C_SDA and I2C_SCL respectively?

Yes. The I2C pins of the ESP32 chip can be remapped by any available GPIO. Please refer to Section 2.2, Pin Overview of ESP32 Series Datasheet. If an external 32.768 kHz crystal is not required, GPIO32 and GPIO33 can be used as I2C pins.

Why does the I2C bus occasionally receive incorrect data when Bluetooth or Wi-Fi is enabled on the ESP32 chip?

This is due to an issue with the I2C FIFO on ESP32, which may cause data confusion when the I2C host reads data. A software fix has been made in release/v5.4 and subsequent versions. For chips released after ESP32 (such as ESP32-S3, ESP32-C3), this issue has been resolved through hardware fixes.

Can the I2C and LP I2C be used simultaneously on ESP32-C6 when it is not in Deep-Sleep mode?

Sure. In the `esp_driver_i2c` driver, you can select `LP_I2C_PORT` to enable the LP I2C function. For details, please refer to example: esp-idf/components/esp_driver_i2c/test_apps/i2c_test_apps/main/test_lp_i2c

---

How to set clock stretching for I2C?

In the `esp_driver_i2c` driver, you can set the `scl_wait_us` parameter in `i2c_master_dev_t` to change the SCL wait time, thereby achieving clock stretching.

Where can I find the I2C timing parameters for the ESP32 series chips?

The Chip Datasheet and Technical Reference Manual do not specify I2C timing parameters. Any tining that complies with I2C timing specifications is considered valid.

Why is there a delay between the 8th and 9th clock during I2C read/write operations on the ESP32 chip?

This is a limitation of ESP32 hardware design, where the hardware needs to handle specific internal logic during this period. Other series of chips do not have this issue.

4.4.5 Inter-IC Sound (I2S)

Does ESP32 support using crystal oscillator as the clock source of I2S?

No. Please go to ESP32 Technical Reference Manual to read about clock source configurations of I2S.

When working as the I2S master, does ESP32 support connection to the I2S slave that only has the three signal lines, I2S_DATA, I2S_SCK, and I2S_WS?

Yes, but the connection to MCLK depends on the requirements of the codec chip on the other side.

Does the I2S interface of ESP32-C3 series chips support the PDM RX mode?

- In the software driver, the I2S interface of ESP32-C3 series chips does not support the PDM RX mode. Unlike ESP32-S3, the PDM RX of ESP32-C3 does not have a module supporting converting from PDM to PCM, which means the acquired data is in the RAW PDM format. The data in this format can't be used directly in most cases.

Can the input and output BCK pin of ESP32-S3 I2S use the same pin?

Yes. In the duplex mode, TX and RX achieve duplex functionality by binding these two signals to the same pin.

Is there a difference between using the Audio PLL (APLL) and PLL in I2S?

APLL and PLL clock sources do not show a significant difference at lower sampling rates (144 kHz and below).

4.4.6 LCD

Where are the LCD drivers and reference examples for ESP series chips?

- Please first understand the LCD development process through the ESP-IoT-Solution Programming Guide - LCD Development Guide.
- For basic LCD driver examples, please refer to esp-idf/examples/peripherals/lcd.
- For advanced LCD driver examples, please refer to esp-iot-solution/examples/display.

Which adapted ICs can be used by the LCD screen of ESP32 series chips?

Please refer to the ESP-IoT-Solution Programming Guide - LCD Development Guide to learn about the adapted LCD screen models.

Common issues in application development with the ESP series chips

For common issues about the LCD screen, please refer to the LCD Application Development Notes.

Notes for Driving MIPI-DSI LCD with ESP32-P4

- ESP32-P4 supports MIPI-DSI LCD with up to 2 lanes. Each supports a maximum rate of 1.5 Gbps, totaling 3 Gbps. It also supports color formats `RGB565`, `RGB666`, and `RGB888`.
- Some MIPI-DSI LCDs, such as `ILI9881` and `JD9365`, are configured as 4-lane by default through hardware (IM[0:3]). However, they can be configured to 2-lane by modifying the LCD' s initialization commands (e.g., the `B6h-B7h` command of ILI9881). Therefore, it is necessary to consult the datasheet of the LCD driver IC to confirm if it is supported.
- The MIPI-DSI driver has the communication response mechanism enabled by default. If there is a communication anomaly between the ESP and the LCD, the ESP may freeze and trigger the watchdog. At this point, please check whether the hardware connection is correct, whether the initialization command is correct, whether the reset timing is correct, or use a logic analyzer to check if the communication is normal.

Do ESP series development boards with screens support GUI development using the Arduino IDE?

- The official LCD driver library for Arduino development, ESP32_Display_Panel, has been released. It can be directly downloaded on the Arduino IDE. For supported development boards, please refer to the documentation.
- Several points to note:
  - ESP32_Display_Panel relies on arduino-esp32.
  - Due to the *screen drift* issue with the RGB interface of ESP32-S3, it is necessary to use features from ESP-IDF release/v5.1 or later to solve the problem. However, the ESP-IDF version used in arduino-esp32 v2.x.x is v4.4.x, which cannot solve this problem. Therefore, you should use arduino-esp32 v3.x.x. For detailed instructions, please refer to Document.
  - Given that Arduino cannot adjust various parameter configurations, such as compile optimization levels, through menuconfig like ESP-IDF, it is recommended to develop a GUI based on ESP-IDF to achieve optimal performance.

How can I improve the display frame rate of LCD screens?

- For a detailed introduction to frame rates, please refer to the ESP-IoT-Solution Programming Guide - LCD Overview. Generally speaking, due to the computational performance of the ESP, the "interface frame rate" is often much higher than the "rendering frame rate", so this issue can be described as "how to improve the rendering frame rate of the LCD". This issue can be considered from the following three aspects:
  - Improve the performance of ESP. When developing with ESP-IDF, ESP can be configured through menuconfig, but it is not configured to the best performance by default. Here, taking `ESP32-S3` as an example, we can increase the CPU frequency to the highest `240 MHz`, increase the frequency of FreeRTOS tick to 1000, and increase the bandwidth of Flash or PSRAM. In addition, we can also increase the data cache line size, set the compilation optimization level to `-O2`, and so on.
  - Improve the performance of LVGL. LVGL itself can also be configured through menuconfig or the *lv_conf.h* file, such as setting LVGL to use the `malloc` and `memcpy` memory operation functions in ESP-IDF, enabling the `fast memory` compilation option, etc.
  - Optimize application design. You can make full use of CPU resources by adjusting the priority of tasks or specifying CPU cores for LVGL and other tasks, especially for ESPs with dual cores. Besides, you can also optimize the design of the GUI, such as avoiding the use of complex animations and layers with transparency overlay effects as much as possible.
- Taking ESP32-S3R8 as an example, the following ESP configuration items can improve the frame rate (ESP-IDF release/v5.1):
  - `CONFIG_FREERTOS_HZ=1000`
  - `CONFIG_ESP_DEFAULT_CPU_FREQ_MHZ_240=y`
  - `CONFIG_ESPTOOLPY_FLASHMODE_QIO=y`
  - `CONFIG_ESPTOOLPY_FLASHFREQ_120M=y` [should be consistent with PSRAM]
  - `CONFIG_SPIRAM_MODE_OCT=y`
  - `CONFIG_IDF_EXPERIMENTAL_FEATURES=y` and `CONFIG_SPIRAM_SPEED_120M=y` [should be consistent with FLASH]
  - `CONFIG_SPIRAM_FETCH_INSTRUCTIONS=y`
  - `CONFIG_SPIRAM_RODATA=y`
  - `CONFIG_ESP32S3_DATA_CACHE_LINE_64B=y`
  - `CONFIG_COMPILER_OPTIMIZATION_PERF=y`
- On ESP32-P4, you can adjust the following configurations to increase the frame rate (applicable to ESP-IDF release/v5.4):
  - `CONFIG_CACHE_L2_CACHE_256KB=y`
  - `CONFIG_CACHE_L2_CACHE_LINE_128B=y`
  - `CONFIG_SPIRAM_XIP_FROM_PSRAM=y`
  - `CONFIG_COMPILER_OPTIMIZATION_PERF=y`
- The following LVGL configuration items can help improve the frame rate (LVGL v8.4):
  - `#define LV_MEM_CUSTOM 1` or `CONFIG_LV_MEM_CUSTOM=y`
  - `#define LV_MEMCPY_MEMSET_STD 1` or `CONFIG_LV_MEMCPY_MEMSET_STD=y`
  - `CONFIG_LV_ATTRIBUTE_FAST_MEM=y`

- If higher frame rates, better rotation performance, and improved anti-tearing behavior are required, it is recommended to use the esp_lvgl_adapter component.

Is there any example code for I2S driving LCD with ESP32?

- The interface type of the screen driven by I2S in ESP32/ESP32-S2 is i80(8080)
- For the application examples, please refer to *LCD examples*.

What is the maximum resolution supported by ESP LCD? What is the corresponding frame rate?

- For the RGB peripheral interfaces of ESP32-S3 and ESP32-P4, due to their hardware limitations, they theoretically support a maximum resolution of `4096 x 1024` (with a maximum of `4096` horizontally and `1024` vertically); for the other peripheral interfaces of the ESP series chips, there is no "maximum" hardware limitation on how much resolution they can support.
- For specific resolution scenarios and corresponding frame rates, please refer to Selection of LCD Driver Chip.

How can I enable PSRAM 120M Octal (DDR) on ESP32-S3R8?

- ESP-IDF v5.1 or later versions are required.
- Enable configuration items through menuconfig: `IDF_EXPERIMENTAL_FEATURES`, `SPIRAM_SPEED_120M`, `SPIRAM_MODE_OCT`.
- The `ESP32-S3-WROOM-1-N16R16V` module currently does not support this feature. If enabled, the chip may freeze upon power-up and then reset.
- **Please note** it is an experimental feature still in testing and may come with the following temperature risks:
  - The chip may not work properly even with ECC enabled when the temperature is above 65°C.
  - Temperature changes may also cause program crashes when accessing PSRAM/flash. For more details, please refer to SPI Flash and External SPI RAM Configuration.

Does ESP32-S3 require an external PSRAM to use the RGB screen?

- In general, yes. RGB screens require the ESP to provide at least one full-screen-sized frame buffer. However, the resolution of RGB screens is usually large, and ESP32-S3's SRAM might not meet this requirement.
- It's not recommended to use a Quad PSRAM due to its relatively low bandwidth, as this could make the PCLK of the RGB LCD cannot be set to the required frequency.
- It's recommended to use an Octal PSRAM and set the clock to 80 MHz or above.

How can I increase the upper limit of PCLK settings on ESP32-S3 while ensuring normal RGB screen display?

- Typically, the upper limit of PCLK settings is constrained by the bandwidth of the PSRAM. Therefore, you need to enhance the PSRAM bandwidth:
  - Use a higher frequency PSRAM clock or a wider PSRAM bus (Octal).
  - Reduce the PSRAM bandwidth occupied by other peripherals like Wi-Fi, flash, etc.
  - Decrease the Data Cache Line Size to 32 Bytes (set to 64 Bytes when using RGB Bounce Buffer mode).

- Enable the Bounce Buffer mode of the RGB driver. The larger the buffer, the better the effect. For usage, please refer to the documentation. Please note that in this mode, the CPU first moves PSRAM data to SRAM, and then the GDMA transfers data to the RGB peripheral. Therefore, it is necessary to enable `CONFIG_ESP32S3_DATA_CACHE_LINE_64B=y`. Otherwise, it may cause the screen to drift.
- Based on limited testing, for Quad PSRAM at 80 MHz, the highest PCLK setting is around 11 MHz; for Octal PSRAM at 80 MHz, the highest PCLK setting is around 22 MHz; for Octal PSRAM at 120 MHz, the highest PCLK setting is around 30 MHz.
- For applications using LVGL, the task of RGB peripheral initialization can be assigned to the same core as the task of LVGL `lv_timer_handler()`. This significantly increases the upper limit of PCLK settings.

Which image decoding formats are supported by the ESP32-S3 series of chips?

- Currently, ESP-IDF only supports the JPEG decoding format. For an application example, please refer to esp-idf/examples/peripherals/lcd/tjpgd.
- If you develop based on LVGL, PNG, BMP, SJPG and GIF decoding formats are supported. For details, please refer to LVGL libs.
- Additionally, if there is a need for animated graphics, you can use the EAF animation format customized by Espressif. For details, refer to the LVGL EAF Player Example.

Why do I get drift (overall drift of the display) when ESP32-S3 is driving an RGB LCD screen?

- **Reasons**
  - The PCLK setting of the RGB peripheral is too high, and the bandwidth of PSRAM or GDMA cannot be satisfied.
  - PSRAM and flash share a set of SPI interfaces. PSRAM is disabled during writes to flash (such as via Wi-Fi, OTA, Bluetooth LE).
  - Reading a large amount of flash/PSRAM data results in insufficient PSRAM bandwidth.
- **Solutions**
  - Improve PSRAM and flash bandwidth. For example, use a higher frequency or larger bit width under the conditions allowed by the hardware.
  - Enable `CONFIG_COMPILER_OPTIMIZATION_PERF`.
  - Reduce the Data Cache Line Size to 32 Bytes (set to 64 Bytes when using the RGB `Bounce Buffer` mode).
  - Enable `CONFIG_SPIRAM_FETCH_INSTRUCTIONS` and `CONFIG_SPIRAM_RODATA`.
  - (Not Recommended) Enable `CONFIG_LCD_RGB_RESTART_IN_VSYNC` to automatically recover after screen drifting, but this cannot completely avoid the issue and may reduce the frame rate.
- **Applications**
  - While ensuring the screen display is normal, try to reduce the frequency of PCLK and decrease the bandwidth utilization of PSRAM.
  - If you need to use Wi-Fi, Bluetooth LE, and continuous flash writing operations, please adopt the `XIP on PSRAM + RGB Bounce buffer` method. Here, `XIP on PSRAM` is used to load the code segment and read-only segment data into PSRAM, and the flash writing operation will not disable PSRAM after it is turned on. `RGB Bounce buffer` is used to block the frame buffer data and transfer it from PSRAM to SRAM through the CPU, and then use GDMA to transfer data to the RGB peripheral. Compared with directly using PSRAM GDMA, it can achieve higher transmission bandwidth. The setup steps are as follows:
    - Make sure the ESP-IDF version is release/v5.0 or newer (released after 2022.12.12), as older versions do not support the `XIP on PSRAM` function. (release/v4.4 supports this function through patching, but it is not recommended)
    - Confirm whether `CONFIG_SPIRAM_FETCH_INSTRUCTIONS` and `CONFIG_SPIRAM_RODATA` can be enabled in the PSRAM configuration. If the read-only data segment is too large (such as a large number of images), it may cause insufficient PSRAM space. At this time, you can use the file system or make the images into a bin to load into the designated partition.

    * Check if there is any memory (SRAM) left, and it takes about [10 * screen_width * 4] bytes.
    * Set `Data cache line size` to 64 Bytes (you can set `Data cache size` to 32 KB to save memory).
    * Set `CONFIG_FREERTOS_HZ` to 1000。
    * If all the above conditions are met, you can refer to the documentation to modify the RGB driver to `Bounce buffer` mode. The `Bounce buffer` mode allocates a block of SRAM memory as an intermediate cache, then quickly transfers the frame buffer data in blocks to SRAM via the CPU, and then transfers the data to the RGB peripheral via GDMA, thus avoiding the issue of PSRAM being disabled. If drift still occurs after enabling, you can try to increase the buffer, but this will consume more SRAM memory.
    * If you still have the drift problem when dealing with Wi-Fi, you can try to turn off `CONFIG_SPIRAM_TRY_ALLOCATE_WIFI_LWIP` in PSRAM, which takes up much SRAM space.
    * The effects of this setting include higher CPU usage, possible interrupt watchdog reset, and higher memory overhead.
    * Since the Bounce Buffer transfers data from PSRAM to SRAM through the CPU in GDMA interrupts, the program should avoid performing operations that disable interrupts for an extended period (such as calling `portENTER_CRITICAL()`), as it can still result in screen drifting.
  - For the drift caused by short-term operations of flash, such as before and after Wi-Fi connection, you can call `esp_lcd_rgb_panel_set_pclk()` before the operation to reduce the PCLK (such as 6 MHz) and delay about 20 ms (the time for RGB to complete one frame), and then increase PCLK to the original level after the operation. This operation may cause the screen to flash blank in a short-term.
  - If unavoidable, you can enable `CONFIG_LCD_RGB_RESTART_IN_VSYNC` or use the `esp_lcd_rgb_panel_restart()` to reset the RGB timing to prevent permanent drifting.
  - For guidance on how to avoid RGB screen drift issues in Arduino, please refer to the link.

Why is there vertical dislocation when I drive SPI/8080 LCD screen to display LVGL?

If you use DMA interrupt to transfer data, `lv_disp_flush_ready()` of LVGL should be called after DMA transfer instead of immediately after calling `draw_bitmap()`.

When I use ESP32-C3 to drive the LCD display through the SPI interface, is it possible to use RTC_CLK as the SPI clock, so that the LCD display can normally display static pictures in Deep-sleep mode?

- Deep-sleep mode: The CPU and most peripherals are powered down, only the RTC memory is operational. For more details, please refer to the "Low Power Management" section in the ESP32-C3 Datasheet.
- The SPI of ESP32-C3 only supports two clock sources, APB_CLK and XTAL_CLK, and does not support RTC_CLK. Therefore, the LCD screen cannot display static pictures in Deep-sleep mode. For details, please refer to *ESP32-C3 Technical Reference Manual* > *Reset and Clock* [PDF].
- For the LCD screen driven by the SPI interface, the driver IC generally has built-in GRAM. Thus, the static pictures can be displayed normally without the ESP continuously outputting the SPI clock, but the pictures cannot be updated during this period.

Are 9-bit bus and 18-bit color depth supported if I use the ILI9488 LCD screen to test the screen example?

The ILI9488 driver chip can support 9-bit bus and 18-bit color depth. However, Espressif's driver can only support 8-bit bus and 16-bit color depth for now.

When using ESP32-S3 to drive an RGB screen, why does it halt or reset (TG1WDT_SYS_RST) when running `esp_lcd_new_rgb_panel()` or `esp_lcd_panel_init()`?

- Please check if the pins occupied by PSRAM in ESP chips or modules conflict with the RGB pins. If there is a conflict, modify the RGB pin configuration.
- If using ESP32-S3R8, avoid using GPIO35, GPIO36, and GPIO37 pins.

When using ESP32-S3 to drive an RGB screen, an abnormal color inversion is observed, i.e., black turns into white, and white turns into black. How to handle this?

Please check whether the initialization register of the screen driver IC has set the invert_color function. For example, in ST7789, this can be corrected by configuring the Inversion register:

- INVOFF (20h): Display Inversion Off
- INVON (21h): Display Inversion On

How to handle color inaccuracies, such as missing colors, when driving an RGB screen with ESP32-S3?

It's likely that the RGB configuration is incorrect. This problem can be troubleshot in the following ways:

- Check for RGB/BGR setting errors: For example, if the screen is set to red (0xC0, 0x0, 0x0), but the screen actually displays black.
- Check whether the RGB and BGR registers are set: For example, in ST7789, it can be corrected through the MADCTL (36h) register (when MADCTL (36h) = 1, it is BGR; when MADCTL (36h) = 0, it is RGB).
- Check for LVGL SWAP16 setting errors: If the screen is configured to red (0xC0, 0x0, 0x0), but the screen actually displays blue, please go to menuconfig → Component config → LVGL configuration → Color settings.
- If there's missing colors in the RGB TTL screen display, it is necessary to set R, G, B displays separately, and check whether the channel with waveform and RGB data line design are compliant.

The spaces in the LVGL's label are correctly inputted, for example "Indoor temperature 25.5℃", but the spaces are not displayed on the screen. What could be the reason and how to troubleshoot this?

This pertains to the missing display of the LVGL label. Enable the following debug items and missing characters will be filled with squares to prevent map loss:

- `Component config → LVGL configuration → Font usage → Enable drawing placeholders when glyph dsc is not found`

When LVGL v8 continuously loads different images stored on flash, the speed is too slow. For example, how to avoid the slow speed issue when cycling through three images on the home screen?

- The reason for the slow speed is that the corresponding image caching mechanism is not turned on, so each images need to be parsed by the parser each time it is used.
- Enable the corresponding image caching mechanism via the `#define LV_IMG_CACHE_DEF_SIZE 1` macro, where 1 represents the number of cached images. Please note that this operation will consume more memory.

LVGL v8 fails to load PNG, JPEG images from flash. What's the reason for a blank screen?

- First, it is necessary to check the status of the remaining memory. LVGL needs to perform two steps to load images: loadpng_get_raw_size and loadpng_convert. If the memory is not enough, it will directly return error code 83.
- The memory requirements should also be estimated in advance: `loadpng_get_raw_size` needs memory equivalent to the image size, `loadpng_convert` requires memory of image length * width * 3 bytes. Enabling the image caching mechanism will cause large `image_cache`, which will simultaneously lead to memory strain.

How to convert a GIF animation into C language code?

Convert the GIF to Map option, with the Color format set to CF_RAW.

Can the screen be set to transparent when displaying GIF animations?

Yes. But GIF only has a 1 bit Alpha descriptor, so it can only be fully transparent or opaque, and there is no semi-transparency.

Which image format is better for the LVGL interface? Is there any reference?

You can refer to the table below:

Image format	Transparent support	Size	Decoding speed
PNG	Perfectly supported	Moderate	Moderate
BMP	Limited support	Large	Fastest, no decoding required
JPG	Not supported	Small	Fast

When converting images to MAP format via imageconverter, if using non-RAW formats such as CF_TRUE_COLOR for conversion, subsequent LVGL loading will not require re-decoding, but it will occupy a larger code segment.

When using some third-party libraries such as FreeType and Lottie with LVGL, why does the screen go blank despite the program loading normally?

First, consider whether the task stack settings are incorrect. Generally, more than 30 KB of task stack needs to be allocated. Please refer to the following demos:

- freetype demo
- lottie porting

What are some good solutions if the internal RAM of ESP32-S3, driving an SPI screen, is insufficient to allocate space for the entire screen buffer?

Use PSRAM as a framebuffer, and then use a small SRAM buffer to transfer data to the framebuffer in multiple batches (SPI DMA can't directly transfer PSRAM data). After completing the transfer, use the framebuffer to render directly. Compared to rendering directly with a small buffer and then sending data, this can prevent tearing and speed up rendering. For specific implementation, please refer to esp_lvgl_port.

How to deal with diagonal tearing on the SPI screen after the hardware is rotated 90 or 270 degrees?

It is recommended to enable the LVGL sw_rotate flag in normal mode to use LVGL's sw_rotate function for software rotation. However, please note that this function conflicts with full_refresh and direct_mode, so do not use them together. For example, calling sw_rotate under full_refresh will directly return without any effect.

Using the ESP32-S2 USB camera and I80 LCD simultaneously may result in the LCD display showing missing images or behaving abnormally. How can this be resolved?

Please refer to this code to increase the startup delay time of I2S.

How to solve the unexpected crash when operating LVGL controls through non-LVGL tasks?

When operating LVGL controls, use `display_lock()` and `display_unlock()` to protect operation variables, thereby ensuring thread safety.

Does ESP32-S3 support RGB888?

Parallel RGB888 is not supported (ESP32-P4 supports parallel RGB888), only RGB565 is supported. Serial RGB888 output can be configured with the following settings:

```
esp_lcd_rgb_panel_config_t panel_conf = {
...
.data_width = 8,
.bits_per_pixel = 24,
...
}
```

How can I disable the left and right swipe functionality when operating the LVGL tabview?

Please add the following code: `lv_obj_clear_flag(lv_tabview_get_content(tabview), LV_OBJ_FLAG_SCROLLABLE);`.

Does LVGL support multiple indev inputs?

Yes. All input devices are managed in a linked list, supporting multiple input devices of the same and different types. For application examples, please refer to the component espressif/esp_lvgl_port, the current component supports inputs such as touch, button, knob, and hid_host.

Does a high CPU usage rate reported by LVGL have any impact?

The CPU usage calculated by LVGL statistics is the duration of the LVGL rendering task within 500 ms, and it does not represent the real CPU usage. Please use FreeRTOS's vTaskGetRunTimeStats to calculate the real usage.

Can ESP32-S3 enter Light-sleep mode after enabling the RGB screen driver?

No, it can't. When initializing the RGB interface, if `CONFIG_PM_ENABLE` is enabled, `ESP_PM_NO_LIGHT_SLEEP` will be automatically locked, preventing the system from entering Light-sleep mode. At this time, please execute `lcd_rgb_panel_destory` to disable the RGB screen driver before entering Light-sleep mode.

Is it supported to drive segment LCD screens?

ESP chips can't directly drive the segment LCD screen through the GPIO pin, because this function requires cycling between high and low voltage levels, with an AC voltage from 2.7 V to 5.0 V and typical values of 3.0 V, 3.3 V, 4.5 V, and 5.0 V. However, the chips do not support voltage range adjustment.

Does ESP32-P4 support HDMI signal output?

ESP32-P4 does not support direct HDMI signal output. However, HDMI signal output can be achieved through an MIPI-DSI to HDMI bridge chip. Currently, the MIPI-DSI to HDMI chip supported by Espressif is LT8912B. For more information, refer to example code.

In LVGL applications, how to rotate the RGB or MIPI-DSI interface screen by 90 degrees/270 degrees? How to avoid tearing? How to further improve the frame rate?

- It is recommended to use the esp_lvgl_adapter component. This component provides features such as high-efficiency rotation, tear prevention, frame rate enhancement, and supports LVGL v8 and v9 versions.
- Refer to the example code esp_lvgl_adapter example.

Can ESP32-P4 generate MIPI-DSI signals (LP and HP) when no MIPI peripheral is connected?

Theoretically possible, but with limitations: Signals can be output during the LP (Low Power) phase; however, during the HP (High Performance) phase, because the driver enables frame ACK by default and has no timeout mechanism, when the FIFO is full it will continuously wait for ACK (BTA), causing the link to freeze. If only capturing waveform segments can satisfy your requirements, then it ' s feasible. If continuous output is needed, you must modify the MIPI-DSI driver inside IDF; currently it cannot be achieved through external configuration alone.

GIF animations in EAF format play slowly on ESP32-S3/P4. How to optimize?

- You can change the EAF encoding format to JPEG (set in the conversion tool), which will increase file size but decode faster;
- For larger resolution screens (such as 466×466), LVGL rendering burden is heavy;
- Compared with EAF + LVGL, using the `esp_emote_gfx` component directly for screen refreshing (segmented decoding) is more efficient. EAF was originally designed for `esp_emote_gfx`, and is only additionally adapted to LVGL;
- AVI with JPEG-encoded frames is smoother on an LCD than GIF;
- P4 revision v3.1 improves CPU frequency and PSRAM bandwidth, enabling LVGL and MP4 to stably reach 15 frames under 1080p RGB888.

4.4.7 LED Control (LEDC)

What is the frequency range for ESP8266 PWM?

The PWM of ESP8266 is realized via software programming, so the maximum CLK value is 1 M limited by timer. It is recommended to set the frequency to 1 K. The PWM frequency can also be improved by decreasing the resolution of duty cycle.

Are there any limits on outputting PWM via ESP32 GPIO pins? Can I distribute it to any I/O?

- The ESP32 can output PWM using any GPIO via IO Matrix. Theoretically, the PWM can be distributed to any I/O except for those that only have input functions (e.g., GPIO34 ~ GPIO39).
- In the actual use, this could also be affected by the limitations of chips and modules, the un-pinned I/Os, flash occupations and etc.

The PWM of ESP8266 NonOS SDK changes slow. What could be the reasons?

- If you are using the gradient APIs in SDK example/IOT_demo, e.g., `light_set_aim` or `light_set_aim_r`, it will need a gradual process for PWM changes.
- If you need the PWM Duty to take effect immediately after configuration, please call API `pwm_set_duty`, and call `pwm_start` next to make this configuration take effect.

When the LEDC is in decremental fade mode, a duty overflow error can occur. How to solve the issue?

When using LEDC, avoid the concurrence of following three cases:

- The LEDC is in decremental fade mode;
- The scale register is set to 1;
- The duty is $2^{\text{LEDC_HSTIMERx_DUTY_RES}}$ or $2^{\text{LEDC_LSTIMERx_DUTY_RES}}$.

When using ESP8266 to generate PWM by directly writing to the register of the hardware timer FRC1, I found there are error PWM outputs after Wi-Fi is initialized since it may disturb the interrupt of FRC1. Is it possible to use FRC2 instead to generate PWM? Or is it possible to set FRC1 higher priority than Wi-Fi?

FRC2 cannot be used as it is occupied by the system. Wi-Fi uses NMI interrupt, which have a higher priority than other ordinary interrupts. It is recommended to use the PWM library of ESP8266_RTOS_SDK. Please refer to ESP8266_RTOS_SDK/examples/peripherals/pwm example.

I' m using v3.3.3 version of ESP-IDF to test the ledc example on ESP32. The LED PWM outputs when Auto Light Sleep mode is disabled, but does not output when this mode is enabled. According the description of LED PWM in ESP-IDF programming guide, LED PWM should work in sleep modes. What is the reason?

v3.3.3 does not support LED PWM working in sleep modes. Please use the LEDC example under the new versions of ESP-IDF (v4.0 and later versions) to test, e.g., ESP-IDF release/v4.2 version of the SDK. Plus, it is also necessary to change the LED PWM clock source to the internal RTC_8M clock source. Please see below:

```
ledc_timer_config_t ledc_timer = {
      .duty_resolution = LEDC_TIMER_13_BIT,
      .freq_hz = 5000,
      .speed_mode = LEDC_LOW_SPEED_MODE,
      .timer_num = LEDC_TIMER_0,
      .clk_cfg = LEDC_USE_RTC8M_CLK,
  };
```

Does ESP32 PWM support complementary outputs with dead bands on two channels?

- This feature is not supported by LEDC but by the MCPWM peripheral.
- By measurement, ESP32-S3 can generate complementary output waveforms with the frequency of 10 k, the duty cycle accuracy of 1 us and the dead band accuracy of 100 ns by MCPWM.

Does LEDC support hardware gamma dimming?

Chips that support the macro `SOC_LEDC_GAMMA_CURVE_FADE_SUPPORTED` can enable hardware gamma dimming by calling `ledc_fill_multi_fade_param_list` and `ledc_set_multi_fade_and_start`.

How to deinit or uninstall ESP32 LEDC pin for use by other peripherals?

To free up the LEDC pin for use by other peripherals, simply call the `ledc_stop` interface. No other AIPs are required.

4.4.8 Motor Control Pulse Width Modulator (MCPWM)

Does ESP32 support using the MCPWM Timer to trigger AD sampling?

AD sampling can be triggered in the callback of the MCPWM timer events `on_ful`, `on_empty`, `on_stop`. Additionally, AD sampling can also be triggered in the callback of the MCPWM comparator event `on_reach`.

Can ESP32-S3 generate fully complementary PWM with accurate clock and duty cycle and adjustable dead band?

By measurement, ESP32-S3 can generate complementary output waveforms with the frequency of 10 k, the duty cycle accuracy of 1 us and the dead band accuracy of 100 ns by MCPWM.

Does the ESP32-S3 support driving eight servos?

Yes. As ESP32-S3 supports two MCPWM controllers, which includes three operators each, and each operator has two generators and two comparators. So, if each generator independently uses a comparator, it can produce a total of 2*3*2=12 PWM outputs with independently adjustable duty cycles, which can naturally meet the needs of eight servos.

What are the advantages of MCPWM compared to LEDC?

MCPWM supports complementary PWM output, center alignment, dead time insertion, carrier modulation, fault, capture events, etc., which are more advantageous in digital motor control, digital power supply, and FOC scenarios.

Why is the PWM frequency output by MCPWM unstable?

The MCPWM does not have a fractional divider, so under frequencies that cannot be divided evenly by the MCPWM clock source, the PWM frequency may become unstable. However, the average frequency remains accurate.

4.4.9 Pulse Counter (PCNT)

Does ESP8266 support pulse counting?

- The ESP8266 does not include a hardware pulse counting module, thus only supports counting via the interrupt of GPIO rising edge or falling edge.
- When Wi-Fi is turned on in ESP8266, it may cause a vacuum in the GPIO sampling due to its high priority, thus interrupting the collected counts and causing data loss.
- In conclusion, it is recommended to use ESP32 and subsequent chips for scenarios with high counting demands.

Can ESP32-S3 realize low-level pulse counting with a frequency of 200 k?

Yes.

Can PCNT generate an interrupt when the count changes?

PCNT will only generate an interrupt when it reaches the set threshold. Other count values can only be read by polling.

What should I do if ESP32-C3 doesn't support PCNT?

The knob software can be used for PCNT counting. Please note that this software can only be used for counting in quadrature decoding related application scenarios, and the counting frequency should not be too high.

Can PCNT support multiple channels working simultaneously? Do signal and control necessarily need to be input in pairs?

PCNT supports multiple channels working simultaneously. The signal and control do not necessarily have to be used in pairs. If you do not want to use control, you can set it to -1. The count mode is configured using *pcnt_channel_set_edge_action*.

Is it possible to use ESP32 PCNT to calculate the time between two pulses?

No, it's not possible. PCNT can only count pulses, it does not support calculating the time between two pulses. You can use RMT to calculate the time between two pulses.

4.4.10 Remote Control Transceiver (RMT)

What can the RMT peripheral on ESP chips be used for in practical?

- For applications of RMT, please refer to RMT application examples. It can be used for infrared remote control, LED strip lighting, D-shot motor control, and so on.

Which ESP chip is recommended for utilizing the RMT functionality?

- ESP32-S3 is recommended because it is currently the only chip with RMT DMA support. This ensures RMT is not interfered by interrupts caused by Wi-Fi, Bluetooth, and other peripherals.

How to change the clock to REF_TICK in RMT?

CHIP: ESP32 | ESP32-S2 | ESP32-C3

By using the rmt_set_source_clk interface.

When ESP32 RMT controls the WS2812 light strip with Wi-Fi or Bluetooth enabled, there are some data frame exceptions. How to solve the issue?

- This problem is difficult to solve on non-ESP32-S3 chips, because the RMT lighting up LED (especially when many LEDs) rely heavily on interrupts, and does not support DMA, thus requiring software to switch ping-pong buffer in the interrupt. If the interrupt does not respond in time, there will be a problem. By default (one memory block is set), after lighting up two LEDs, the RMT driver will go into the interrupt to switch the internal ping-pong buffer.
- Workaround:
  - For esp-idf release/v4.4 and earlier versions, increase mem_block_num. There is a change in release/v5.0. Please see Breaking Changes in Usage。
  - Install the RMT interrupts on a specific CPU core by calling the driver install function in a pin-to-core task, thus avoiding the core used by Wi-Fi or Bluetooth.
  - You can also use SPI DMA as an alternative to RMT to solve this issue. For more details, please refer to the SPI DMA LED strip example.
- If you are in the product selection stage, it is recommended to use ESP32-S3 RMT.

How can I quickly adapt other infrared protocols based on the IR NEC example in ESP-IDF ?

- You can accelerate the adaptation of other infrared protocols based on the IR NEC example by using the RMT Encoder.
- If you need the infrared learning function, you can use the ir_learn component.

ESP32-S3 RMT supports configuring 4 RMT RX/TX channels. Why does it fail when creating more than 2 RMT TX channels in a row using rmt_new_tx_channel?

- This is because the *mem_block_symbols* parameter configured in the *tx_chan_config* structure is too large. On ESP32-S3, the size of each dedicated memory block for RMT is 48 bytes. If the *mem_block_symbols* parameter exceeds 48, creating a TX channel will actually occupy the memory block of the next adjacent channel as well. Therefore, if you want to create and use 4 RMT RX/TX channels simultaneously, the *mem_block_symbols* parameter should not exceed 48.
- Please note that the size of each dedicated memory block for RMT on ESP32 is 64 bytes.

Can ESP32-S3 RMT achieve synchronized output for multiple TX channels?

- Yes, please refer to the following code snippet:

```
rmt_channel_handle_t tx_channels[TEST_RMT_CHANS];
rmt_sync_manager_handle_t synchro = NULL;
rmt_sync_manager_config_t synchro_config = {
  .tx_channel_array = tx_channels,
  .array_size = TEST_RMT_CHANS,
};
rmt_new_sync_manager(&synchro_config, &synchro);
for (int i = 0; i < TEST_RMT_CHANS; i++) {
  rmt_transmit(tx_channels[i], led_strip_encoders[i], leds_grb, TEST_LED_
↪NUM * 3, &transmit_config);
}
```

How can I achieve cyclic data transmission using the RMT TX channel on ESP32-S3, such as an infinite loop?

- You can realize infinite loop transmission by setting the *rmt_transmit_config_t::loop_count* to -1. For more details, please refer to Initiate TX Transaction。

Does the ESP32-S3 support One-Wire?

- ESP32-S3 can support the One-Wire bus protocol through the RMT peripheral. For specific applications, refer to the "esp-idf/examples/peripherals/rmt/onewire" example.

Can the `rmt_transmit` function be used in an interrupt?

No, the `rmt_transmit` function can only be called in a task environment, not in an interrupt.

4.4.11 Secure Digital Input Output (SDIO)

What is the maximum speed supported by the SDIO interface?

- The maximum clock speed supported by ESP32 SDIO is 50 MHz, and ESP32 SDIO supports the Quad mode at the maximum.
- The maximum clock speed supported by ESP32-S3 SDIO is 80 MHz, and ESP32-S3 SDIO supports the Octal mode at the maximum.
- The practical speed is influenced by the read and write speed of storage media at the same time.

Does the ESP8266 SDIO support connecting to an SD card?

ESP8266 only has an SDIO slave interface, so it does not support connecting to an SD card. If you need an ESP chip to connect to an SD card, please choose one that supports the SD/MMC host interface, such as the ESP32-S3.

What is the maximum capacity for ESP32 SD card?

- In the SD3.01 Specifications, the SDXC card supports a maximum capacity of 2 TB (2048 GB).
- The ESP32 SDMMC Host also complies with the SD3.01 Specifications, which means up to 2 TB areas of it can be accessed by peripherals. When accessing the card via SPI bus using the SDSPI driver, there are also 2 TB of areas can be accessed in hardware level.
- In software level, the usable area of the card is also affected by the file system.

Is it possible to use ESP32 SD card together with flash & PSRAM?

- Yes, they can be used simultaneously when they apply different pins.
- Note that when ESP32 uses the SDMMC host driver, the SDMMC Slot0 pins in the ESP32-WROOM and ESP32-WROVER modules conflict with the flash.

Does ESP-WROOM-S2 module support using SDIO as a slave?

Yes, it does.

Since ESP32-S2 has removed the SDIO interface, does it still support external TF card?

You can use the SPI2 or SPI3 interface to connect to an external TF card. When doing so, please use the SPI mode of the TF card.

Does ESP32-S2 support eMMC?

CHIP: ESP32-S2

No.

Does ESP32-S2 support SDIO as a slave?

ESP32-S2 has no SDIO interface and does not support SDIO as a slave.

How does ESP32 enable and disable the interrupt for the SDIO slave to receive data?

The data reception of the SDIO slave is related to the state of the mounted buffer. After the data is received, you need to call sdio_slave_recv_load_buf to release the buffer. Otherwise, the SDIO host will not be able to continue to send data to the SDIO slave.

How to set the ESP32-C6 SDIO clock?

The ESP32-C6 only supports SDIO slave mode, with the SDIO slave CLK dependent on the SDIO host settings.

4.4.12 Serial Peripheral Interface (SPI)

Is ESP-WROOM-02D module able to connect SPI flash?

The ESP-WROOM-02D module is a Wi-Fi module based on the ESP8266 chip, which supports communication with external SPI flash devices using SPI interfaces. Specifically, the ESP-WROOM-02D module provides 4 SPI interface pins, GPIO12, GPIO13, GPIO14, and GPIO15. Among these pins, GPIO12~GPIO14 can be used as the MISO, MOSI, and SCLK pins for the SPI master interface, and GPIO15 can be used as the CS pin for the SPI slave interface.

To connect an external SPI flash device to the ESP-WROOM-02D module, the MOSI, MISO, SCK, and CS pins of the SPI flash device should be connected to the GPIO12~GPIO14 and GPIO15 pins of the ESP-WROOM-02D module. Additionally, the SPI interface needs to be properly configured and initialized in the firmware to ensure correct communication of ESP8266 with the external SPI flash device.

It should be noted that the model and capacity of the external SPI flash device should be selected based on the specific application requirements. In addition, timing characteristics and reliability of the SPI flash device should be considered to ensure data can be transmitted correctly and stably. Moreover, factors such as environment noise and physical distance between the SPI flash device and the ESP-WROOM-02D module should be considered to improve the reliability and performance of the system as much as possible.

Taking ESP-WROOM-S2 as the slave device and STM32 as MCU, is it possible to download through SPI interface?

No, we use UART0 to download by default. You can also design OTA support yourself in firmware.

What is the difference among SPI0, SPI1, HSPI and VSPI in ESP32?

- ESP32 has 4 SPIs, SPI0 and SPI1 are two peripherals, also known as MSPI. SPI0 and SPI1 share the same SPI bus (same signals and IOs). The difference is, MSPI CS0 is connected to the main flash for firmware storage, and MSPI CS1 is connected to PSRAM. SPI2 and SPI3 are general-purpose SPIs that are available for customers to use.
- HSPI represents the above-mentioned SPI2, and VSPI represents the SPI3. The two sets of SPIs are general-purpose SPIs and support QSPI.

The maximum data transmission of SPI DMA on ESP series chips is 4095 bytes. Is it because of hardware limitation?

- Yes, this is a hardware limitation. A single node in the DMA table can only mount 4095 bytes of data.
- However, a single SPI DMA transmission can mount more data through several nodes. At this time the maximum data transfer byte number is limited by the hardware register `SPI_LL_DATA_MAX_BIT_LEN` (whose value varies with different chip series and can be found in ESP-IDF), i.e., `max_transfer_sz <= (SPI_LL_DATA_MAX_BIT_LEN / 8)`.

Does the SPI of the ESP series chip need to perform semaphore synchronization to access three SPI slave devices simultaneously?

- The same SPI peripheral, as the master, can only communicate with one slave at a time, and CS decides which slave to communicate with. If you connect 3 slave devices to the SPI driver and communicate with them separately, it is okay and recommended.
- It is recommended to operate devices sharing the same SPI in only one task. Otherwise, it is not thread-safe. Communication needs to be synchronized through semaphores. For specific issues, see SPI Master Driver > Driver Features.

When using an ESP32 board for development and testing based on ESP-IDF release/v4.3, I received the following error log during compilation. What is the reason?

```
spi_flash:Detected size(8192K) smaller than the size in the binary image␣
↪header(16384K).Probe failed.
```

The reason is that the configured flash size is larger than the actual flash size. In order to avoid misuse of a larger address space, the actual flash size is checked.

What is the maximum transmission speed supported by SPI slave?

CHIP: ESP32

ESP32 can support up to 10 M of transmission speed when serves as an SPI slave.

How many bytes of data can be transferred at once using the ESP series chip as an SPI host device in non-DMA mode?

- Due to the limitations of the SPI hardware FIFO, a maximum of 64 bytes can be transferred at once in non-DMA mode. Refer to SPI Master Driver > Transaction Duration.
- When transmitting no more than 32 bits, the 4-byte array inside the SPI Master driver can be used as the buffer for sending data. Refer to SPI Master Driver > Transactions with Data Not Exceeding 32 Bits for details.
- When transmitting more than 32 bits, you need to create an SPI data transmission buffer yourself. You can refer to the SPI Master Driver > SPI Transactions for instructions.

When using the ESP32-S3-WROOM-1 (ESP32-S3R2) module to enable its PSRAM configuration based on the "hello-world" example in ESP-IDF v4.4, the following error is printed. What is the reason?

```
E (232) spiram: Virtual address not enough for PSRAM!
```

ESP32-S3R2 chip integrates a 4-wire 2 MB PSRAM, please set PSRAM Mode to **Quad** mode in menuconfig before your action as follows:

```
menuconfig > Component config > ESP32S3 Specific > Support for external,
SPI connected RAM > SPI RAM config > Mode (QUAD/OCT) of SPI RAM chip
in use (Ouad Mode PSRAM)
```

When using the ESP32-S3-WROOM-2 (ESP32-S3R8V) module to enable the PSRAM configuration based on the "hello-world" example in ESP-IDF v4.4, the following error is printed. What is the reason?

```
E (453) psram: psram ID read error: 0x00ffff
E (454) cpu start: Failed to init external RAM!
```

ESP32-S3R8V chip integrates a 8-wire 8 MB PSRAM, please set PSRAM mode to **Octal** mode in menuconfig before your action as follows:

```
menuconfig > Component config > ESP32S3 Specific > Support for
external, SPI connected RAM > SPI RAM config > Mode (QUAD/OCT)
of SPI RAM chip in use (Octal Mode PSRAM)
```

Does ESP8266 RTOS SDK support full duplex for SPI?

CHIP: ESP8266

No, it doesn't. Because ESP8266 doesn't support DMA, in order to improve the transmission performance, the entire FIFO is used. So it can only be half duplex. Please refer to spi readme for more details.

Can the ESP series chips support the 9-bit clock mode of three-wire SPI (i.e., the mode where the first bit indicates whether the following 8 bits are command or data)?

- Currently, the ESP32, ESP32-S, ESP32-C series chips do not support non-byte aligned data transmission, i.e., they only support 8-bit aligned data transmission. For a detailed explanation of this issue, see Github issue and documentation.
- Subsequent new versions of the ESP chip may support non-byte aligned data transmission, but there is currently no specific timetable.

After setting a pin of the ESP series chip as the SDA data line, the expected result is that the SDA line should be low when idle, and high when writing data. But why is this pin high when idle upon power-up, and low when writing data? How can I achieve the expected result?

- In SPI, the idle levels of the MOSI (SDA) and SCK signal lines are controlled by the SPI mode.
- This can be achieved by modifying the *mode* member variable in the *spi_device_interface_config_t* structure https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/spi_master.html#_CPPv4N29spi_device_interface_config_t4modeE.

How many bytes can a single DMA Buffer support when the ESP32-C6 is in SPI DMA mode?

- For ESP32-C6 in SPI DMA mode, a single DMA Buffer supports 4092 bytes at the maximum.

Does ESP32, as an SPI host, support a 30 MHz clock?

- Not supported. When ESP32 acts as an SPI host, it can support up to 80 MHz CLK when using SPI IO_MUX pins, and supports integer division at 80 MHz.
- When using GPIO matrix pins, it can support up to 40 MHz CLK. If using the full-duplex transmission mode of GPIO matrix pins, it only supports up to 26 MHz CLK. For detailed instructions, refer to the SPI Master Driver software usage instructions.

Why is there a difference in the MISO/MOSI status of the ESP32 and subsequent chips in SPI half-duplex mode?

In half-duplex TX mode, the ESP32 by default pulls the unused MISO line to a low level, while other chips except for the ESP32 default to a high level, which is a normal phenomenon. If you need to pull other chips except for the ESP32 to a low level, you can set `SPI_Q_POL` to 0.

Is the Handshake pin mandatory when using an ESP series product as an SPI slave device?

- The Handshake pin is not mandatory; whether it is required depends on the specific application. In the following official solutions, the Handshake pin is part of the protocol design and is a required signal:
  - SPI AT Solution for ESP-AT (ESP32-C2/C3/C5/C6/C61, etc.): In order to allow Espressif devices, which act as SPI slaves, to actively report data, an additional Handshake line is used. The slave pulls this signal high to notify the master that data is available for reception or that data transmission is permitted. Therefore, the Handshake pin is required in the SPI AT protocol.
  - ESP-Hosted/ESP-Hosted-MCU SPI mode: The protocol explicitly requires additional GPIO connections such as Handshake, Data Ready, and Reset/EN, in addition to the standard Quad SPI. Among them, the Handshake pin is used to indicate that the Espressif peripheral is ready for an SPI transaction, and is a required signal. For more details, please refer to the ESP-Hosted SPI FD (Full Duplex) Operation and SPI Communication Protocol documentation.
- Additionally, the official ESP-IDF spi_slave example also defaults to using a single Handshake line for synchronous control.
- If you are implementing an SPI slave protocol using an ESP product, from a hardware and driver perspective, SPI itself only requires four signal lines: SCLK, MOSI, MISO, and CS. Whether to add a Handshake line depends on the design of the upper-layer protocol and synchronization requirements. For example, the official ESP-IDF spi_slave_hd example does not require a Handshake line.

When using the ESP32-C3 as an SPI slave with the receiver example, all data is incorrect when the master MCU clock is 20 MHz, and there is severe packet loss at 10 MHz. How to solve this problem?

The ESP32-C3 SPI slave receiver example requires handshake line control (the handshake line being pulled high indicates that the DMA is properly mounted). The handshake protocol must be correctly implemented for normal data transmission and reception. If you do not want to use a physical handshake line, you can switch to the `spi_slave_hd` half-duplex scheme, which has an internal mechanism for cyclically mounting DMA buffers, resulting in more stable transmission rates.

4.4.13 Timer

What should I pay attention to when using the HW timer interrupt with ESP8266?

- Please refer to ESP8266 Technical Reference Manual regarding the related APIs.
- If you are using NonOS SDK, please refer to ESP8266 Non-OS SDK API Reference.
- Typically, the hardware interrupt callback function needs to be executed as quickly as possible, and the callback function should be placed in IRAM to avoid the impact of Cache being turned off.
  - RTOS SDK requires adding the linker attribute IRAM_ATTR before the function name.
  - NonOS SDK cannot add ICACHE_FLASH_ATTR (this attribute specifies the function is placed in flash) before the function name.

How to set interrupt priority for timers?

- The esp_timer, using ESP32 as an example, allows you to configure the interrupt priority by modifying the configuration item CONFIG_ESP_TIMER_INTERRUPT_LEVEL in Menuconfig.
- The General Purpose Timer allows setting interrupt priority when registering the interrupt service function. For details, please refer to the API description of timer_isr_callback_add.

How to improve the timeliness of interrupt response in gptimer?

Put the interrupt function and the corresponding callback into the IRAM.

4.4.14 Touch Sensor

Can ESP32/ESP32-S2/ESP32-S3 Touch pass the EMS conduction interference test?

The EMS conduction interference test is a whole machine test, which is jointly affected by internal and external power supplies, circuits, and external environments. Since the ESP32/ESP32-S2/ESP32-S3 Touch has poor anti-interference ability, if users have this test requirement, it is recommended to externally connect a dedicated Touch IC or use ESP32-P4.

Can ESP32-P4 Touch pass the EMS conduction interference test?

Yes, ESP32-P4 Touch can pass the EMS 3V conduction interference test (This requires a dedicated software driver. Please contact Espressif if needed).

When using ESP32 to develop Touch Sensor applications, where can I find references?

Please refer to Software and Hardware Designs.

When there is water on ESP32-S2 Touch Sensor, does it block or recognize the Touch event with its waterproof function?

When there is a small amount of water droplets, the waterproof function of the ESP32-S2 Touch Sensor can work normally. In the event of a large area of standing water (i.e. the Touch contact area is completely covered), the Touch will temporarily lock and will not resume operation until the water is cleared.

While the waterproof feature of ESP32-S2 Touch Sensor shielding the Touchpad with water flow, does other pads with no water still usable?

Yes, the specific shielding channel can be selected via software.

Are there any recommendations for materials that can be used to test Touch Sensor, can trigger Touch Sensor stably and is close to the parameters of human touches?

For experiments with high consistency requirements, it is doable to replace human hands with cell phone pencils.

Can the pins of Touch Sensor be remapped?

No, because Touch Sensor is realized via software programming.

Do I need to reset a check threshold for Touch Sensor after covering it with a acrylic plate?

Yes.

Is it possible for Touch Sensor to detect whether there is a acrylic plate on the top, so that it can switch to the pre-defined threshold value automatically when there is a acrylic plate added or removed?

For now, it cannot adapt to the impacts brought by physical changes.

What reference drivers are available for ESP32 touch screens?

For touch screen drivers, refer to:

- esp-iot-solution touch driver: lcd_touch.
- esp-bsp touch driver: lcd_touch.

How long does it take from the moment of touch to the time when the chip detects the touch and issues a command signal?

If n channels are used, and each channel takes about 2 ms to measure, one round of polling would take 2n ms.

When the ESP32-S3 touch pin is grounded and touched with human body multiple times, the four pin values previously defined changed to a fixed constant, resulting in invalid pin state. What is the reason for this?

The touch pin must not be grounded. This is what happens when it is grounded.

4.4.15 Two-Wire Automotive Interface (TWAI)

What are the considerations when using the ESP32 TWAI® controller?

Please refer to the ESP32 Series SoC Errata > Section *ESP32 TWAI Errata*.

Does ESP32-S3 support CAN-FD?

The ESP32-S3 itself does not integrate a CAN-FD controller, but users can still use a CAN-FD controller with an SPI interface, such as the MCP2518FD. For subsequent chips like ESP32-C5 support CAN-FD. For more information please refer to esp_driver_twai.

Does ESP32 support interrupt mode for CAN reception?

Yes, it does. The TWAI driver uses receive interrupt, and automatically stores the received information into the RX queue.

4.4.16 Universal Asynchronous Receiver/Transmitter (UART)

When using ESP8266 RTOS SDK v2.1 and previous versions, how to set log to UART1?

After initializing UART1, you can switch log to UART1 via API:

```
UART_SetPrintPort(UART1);
```

When using ESP8266 RTOS SDK v3.0 and later versions, how to set log to UART1?

Go to `menuconfig` -> `Component config` -> `ESP8266-specific` -> `UART for console output` -> `custom` -> `UART peripheral to use for console output` -> `UART0` and change the option to "UART1".

How to enable UART Flow Control in ESP32 IDF?

- To enable hardware flow control, please refer to uart-flow-control.
- To enable software flow control, please refer to software-flow-control.

When using UART0 as a serial communication port for ESP32, what should I pay attention to?

- Generally, it is not recommended to use UART0 as a normal serial communication port, because it is the default LOG output port.
- If the UART number in ESP32 is not enough for you or it is not convenient to change your hardware designs anymore, and UART0 is therefore going to be used as a normal communication port, please pay attention to the following suggestions:

Software: You need to protect the serial communication port from being affected by printing. The UART0 mainly has three print settings in the default program:

- The first instance is the Level 1 bootloader (ROM firmware) printout. When powered on, the MTDO pin can be set to a low level to block the ROM firmware printout.
- The second part is the secondary bootloader log output. You can set `menuconfig` -> `Bootloader config` -> `Bootloader log verbosity` to `No output` to block the bootloader log output.
- Third, app log output. You can set `menuconfig` -> `Component config` -> `Log output` -> `Default log verbosity` as `Not output` to block app log output.

Hardware:

- Pay attention to other devices on UART0 when downloading programs since they could affect downloading. It is recommended to reserve a 0 Ω resistance between ESP32 and other devices so that if there is something wrong while downloading, you can still disconnect this resistance.

Is it possible to use GPIO34 ~ GPIO39 from ESP32 as the RX signal pin for UART and TWAI®?

Yes, GPIO34 ~ GPIO39 are for receive only and can be used as the RX signal pins for UART and TWAI®.

How to dynamically change the serial baud rate and make it take effect immediately with ESP32?

You can use the `uart_set_baudrate()` API to dynamically change the UART baud rate. See API Description.

Does the ESP32 chip support USRAT (Universal Synchronous Asynchronous Receiver Transmitter)?

It’s not support. ESP32 only supports UART and cannot provide the synchronous clock.

Does the serial port verification of the ESP32 chip support MARK and SPACE verification?

No.

What is the size of the hardware FIFO in ESP8266’s serial port?

Both UART0 and UART1 of ESP8266 have a 128-byte hardware FIFO and a 128-byte RW FIFO, which operate at the same address. Please refer to Section 11.2. Hardware Resources in ESP8266 Technical Reference Manual.

What is the serial port baud rate range of ESP8266?

300 ~ 115200*40 bps. Please refer to Section 11.3.1. Baud Rate in ESP8266 Technical Reference Manual.

How to modify the output port of UART0?

CHIP: ESP32 | ESP32 | ESP32-C3

This can be set in menuconfig: `idf.py menuconfig` —> `Component config` —> `Common ESP-related` -> `Channel for console output (custom UART)`.

When using ESP8266, I want to use UART0 exclusively for downloading, and then use UART1 to communicate with other chips. Can GPIO4 and GPIO5 be configured as UART1 serial ports?

- Since the RXD of UART1 is occupied, UART1 cannot be used to communicate with other chips, but the TXD pin of UART1 can be used to output logs.
- ESP8266 can only communicate with other chips by swapping CTS and RTS pins of UART0. It will be invalid to configure GPIO4 and GPIO5.
- ESP8266 can communicate with other chips by calling the `uart_enable_swap()` function, swapping the CTS and RTS pins of UART0 for MTCK (IO13) and MTDO (IO15) pins. After the pin swap, ESP8266 can use GPIO13 (RXD) and GPIO15 (TXD) to communicate with other chips via UART.

Can ESP32’s UART0 be used for inputting from the computer console while it is being used for outputting logs?

Yes. Outputting logs only requires using the TXD0 pin, while receiving input from the computer console only requires using the RXD0 pin. You can use the esp-idf/examples/system/console/basic example for testing.

How to realize UART signal inversion?

You can use the uart_set_line_inverse interface to set it. The inverse pin parameter can be obtained from the uart_signal_inv_t variable.

Does the ESP module UART support LIN mode?

LIN mode is not supported at the hardware level. You may conduct simulation on the software side to realize this feature, and there is currently no reference material available from the official source.

Does ESP32 support the UART IRDA working mode?

ESP32 supports the UART IRDA working mode at the hardware level, but there is no corresponding driver implementation in software yet.

In situations where serial port resources are limited, is GPIO-based UART simulation supported?

You can test this using the soft_uart example.

Can UART and LP UART be used simultaneously when ESP32-C6 is not in DeepSleep mode?

Yes, UART and LP UART can be used simultaneously. You can enable the LP Core to use LP UART. Please refer to this use case: esp-idf/examples/system/ulp/lp_core/lp_uart.

Does UART support DMA functionality?

Please refer to UART DMA (UHCI). Please note that UART DMA shares HCI hardware with BT, so do not use BT HCI and UART DMA at the same time.

What is the UART RX sampling mechanism on the ESP32-S3?

Midpoint sampling.

4.4.17 USB

Does ESP32 support USB function?

- No, ESP32 does not support USB function.
- However, ESP32-S2/S3 supports USB2.0 Full-speed mode.
- ESP32-P4 supports USB2.0 High-speed and Full-speed modes.

Does the ESP-IDF SDK USB interface support HID and MSC modes?

- ESP32S2/S3 can be used as MSC Host to support reading from or writing to storage devices such as USB flash disks. For details, please refer to esp-idf.
- ESP32S2/S3 can be used as MSC Device to simulate storage of USB flash disks. For details, please refer to esp-iot-solution.
- ESP32S2/S3 can be used as HID Host. For details, please refer to ESP-IDF Host HID.
- ESP32S2/S3 can be used as HID Device. For details, please refer to ESP-IDF Device HID.

What is the stable current output for ESP32-S2' s USB interface?

The current output capability of the VBUS power line is determined by the power supply, not by the ESP32-S2 chip. If the chip is self-powered, please refer to Self-Powered Device.

Does ESP32-S3' s USB peripheral supports USB Host?

Yes, regarding this function, ESP32-S3 is the same as ESP32-S2.

Does ESP32-C3 USB support USB serial port function and USB JTAG function?

Yes, but you cannot define the descriptor by yourself.

What are the USB features of ESP32-S2 and ESP32-S3?

ESP32-S3 and ESP32-S2 support USB 2.0 OTG (supporting full-speed mode), and both support Host and Device functions. On top of that, ESP32-S3 also supports USB-Serial-JTAG peripheral, which can be used to download and debug firmware.

Are there any references to the library and demo of ESP32-S2 USB Host?

Please refer to USB Host in ESP-IDF.

The USB protocol supported by ESP32-S2 is OTG 1.1, with the maximum speed of 12 Mbps. Can it communicate with USB 2.0 devices?

In the full speed mode, USB 2.0 devices are compatible with USB 1.1 devices, so they can communicate with each other.

Does ESP32-S2/S3 support USB cameras?

Supported. For ESP32-S2/ESP32-S3 USB Host UVC example code, please refer to usb_hub_dual_camera. This example can connect to a single USB camera, or multiple USB cameras through a HUB.

Does ESP32-S3 support USB cameras with microphones and speakers?

Supported. Please refer to the example codes for ESP32-S2/ESP32-S3 respectively:

- Audio: audio_player.
- Camera: basic_uvc_stream.
- Web preview camera: usb_hub_dual_camera.

Is there any reference for the example of using ESP32S2 as a USB flash drive (MSC DEVICE)?

Please refer to usb_msc_wireless_disk demo. The average read and write speed currently tested is: read 540 KB/s, write 350 KB/s.

As ESP32-C3 already has USB function, can I download firmware directly via USB without using the cp2102 chip?

Yes, ESP32-C3 can download firmware via USB, The USB serial port number should be displayed as COMx on Windows devices and ttyACMx on Linux devices.

Does ESP32-C3 support USB Host?

No, it only supports USB-Serial-JTAG function and can only be used as the USB device.

The ESP32-C3 chip can use USB to download firmware, but it is not supported under ESP-IDF v4.3. How to use USB to download firmware?

You need to compile under ESP-IDF v4.4 or later versions. After pulling the latest branch and updating the IDF tool, you can compile normally and download it using USB. Please refer to usb-serial-jtag-console for the usage.

Does the ESP32-S2 support USB HID?

Yes. For the example of USB HID Device, please refer to ESP-IDF Device HID. For the example of USB HID Host, please refer to ESP-IDF Host HID example.

Why is this error log printed when I am testing the USB Camera + Wi-Fi Transfer example?

```
E (1437) UVC STREAM: Configuration descriptor larger than control␣
↪transfer max length
```

This error log is reported because the length of the descriptor sent by the USB Camera is larger than the default length (256). You can modify the following configuration to 2048 for testing:

`Component config > UVC Stream > (2048) Max control transfer data size (Bytes)`

Does ESP32-S3 support USB CDC for printing program log and downloading firmware?

Yes, ESP32-S3 supports printing program log and downloading firmware using USB CDC when the following configuration option is enabled:

`Component config > ESP System Settings > Channel for console output > USB CDC`

Does ESP32-S3 support devices with USB Device being Class 0?

Supported, you can refer to the example: esp-idf/components/tinyusb/additions/src/usb_descriptors.c. When Class code == 00H, the class category is specified by the interface.

Can the ESP32-S3' s USB OTG interface be used in both USB Host and USB Device modes?

- The ESP32-S3 ' s USB OTG interface can not be used as USB Host and USB Device at the same time. However, it is possible to switch between the USB Host mode and the USB Device mode by software.
- If you need the standard negotiation function of USB OTG, please note that currently ESP32-S3 only supports this function on the hardware, and does not support it in software protocol.

When testing the esp-idf/examples/peripherals/usb/device/tusb_serial_device example to send data using TinyUSB, do I have to use the tinyusb_cdcacm_write_flush() function?

To prevent sending FIFO overflows, you can use the 'tinyusb_cdcacm_write_flush()' function to flush. However, a large number of cycles of flushing may fail. So, it is recommended to set it according to the actual application.

Can ESP32-S3 use an external USB hub chip with two of its USB ports connecting to a USB 4G module and a dongle at the same time?

Supported. Two devices can be connected to the ESP32-S3 chip via a USB hub. For the implementation of the USB 4G module, please refer to *4G Internet Solution*; for the encryption dog, you need to develop the corresponding USB Host driver according to the device type.

When ESP32-S2/ESP32-S3 serves as the UVC Host and connects some models of UVC cameras, why is there an error `HID_PIPI_EVENT_ERROR_OVERFLOW` in the log?

This is because MPS of the Alt interface endpoint of the selected camera is too large (ESP32-S2/ESP32-S3 supports up to 512 bytes). Please confirm whether the camera has an interface of less than or equal to 512 bytes under USB1.1.

Does ESP32-S2/ESP32-S3 support USB 4G Internet access solutions?

Yes, depending on the support of the 4G module, refer to (ECM/RNDIS recommended):

- PPP Protocol: USB CDC 4G Module Example
- ECM Protocol: USB ECM 4G Module Example
- RNDIS Protocol: USB RNDIS 4G Module Example

Why can't I access the internet or successfully ping after obtaining an IP using ECM/RNDIS to connect to the 4G module?

- Check if the auto-dial function in ECM/RNDIS mode has been activated. For details, please refer to the corresponding module's datasheet.
- Check if the SIM card is properly inserted and whether it has been successfully registered to the network.

What is the reason for the 4G module printing `No ECM interface found for device VID: xxxx, PID: xxxx, ignore it` after connecting to USB?

The reason is that the ECM interface was not recognized. Please check whether the module has been correctly set to ECM mode. For details, please refer to the documentation of the corresponding module.

What is the reason for the `E (887) HCD DWC: EP MPS (512) exceeds supported limit (408)` printout after connecting the 4G module to USB?

The USB endpoint MPS of this module exceeds the size allowed by the ESP chip hardware, please try:

- Connect using the RNDIS protocol.
- Replace with another 4G module.
- Using the ESP32-P4 chip, this chip supports a larger USB endpoint MPS.

Does it support connecting a 4G module and a USB sound card simultaneously using a USB HUB?

Supported. It is recommended to use 4G modules with RNDIS/ECM protocol and USB UAC sound cards. Refer to *4G Internet Solution* and USB UAC Sound Card Example respectively.

Is there a limit to the number of devices that can be connected via a USB HUB?

There are hardware limitations. The ESP32-S2/ESP32-S3 USB Host peripheral has 8 channels; the ESP32-P4 USB Host HS peripheral has 16 channels; the ESP32-P4 USB Host FS peripheral has 8 channels. One endpoint of a device will occupy one channel, and the HUB itself will also occupy a certain number of channels. Therefore, the actual number of connectable USB devices depends on the number of device endpoints and the number of channels occupied by the HUB.

Is there any USB CDC Host example for ESP32-S2/ESP32-S3?

Yes, please refer to the ESP-IDF USB CDC Host Example or the esp-iot-solution USB CDC Host Example.

When burning firmware through the ESP32-C3/ESP32-S3 USB Serial/JTAG Controller function, I found that the PC sometimes cannot recognize the USB serial port, or automatically disconnects from the USB serial port repeatedly after recognizing it. What is the reason?

At present, the startup logic of the ESP32/ESP32-S2/ESP32-S3/ESP32-C3 chip is: if it cannot start normally (due to empty flash, no correct data/firmware in flash, the power-on sequence problem of flash, etc.), the internal timer will trigger the chip to restart every few seconds. The chip cannot connect stably until the program starts normally or enters the download mode. The ESP32-S3/ESP32-C3 USB-Serial-JTAG peripheral will be re-initialized when the chip restarts, so the corresponding result is that the chip tries to connect to and disconnects from the PC every few seconds. We provide the following two solutions:

- You need to boot the chip to enter the download mode manually before the first download or after flash is erased, so that the chip can be connected stably.
- You can also burn the firmware that can run stably through UART in advance to solve this issue. If there is firmware that can run stably in the chip, the USB serial port of the chip can be connected stably in subsequent programming.

If there is no strap pin test point reserved for booting manually, you may need to try several times in the initial USB download.

Why does ESP32-S2/ESP32-S3 not reach the maximum USB full speed, 12 Mbps?

We will explain this issue with the TinyUSB protocol stack as an example. As this USB mode does not use DMA, but directly uses CPU polling, some time slices are wasted in each transfer. As a result, the TinyUSB protocol stack is only expected to reach 6.4 Mbps (it can reach 9.628 Mbps theoretically if the batch transfer is adopted).

How can I confirm if ESP32-S2/ESP32-S3 USB supports a certain USB camera or not?

ESP32-S2/ESP32-S3 USB only supports USB cameras that correspond to wMaxPacketSize Video Streaming endpoints which include 512 bytes at the maximum. You can use USB Stream Example to test. An error log will be printed if the camera is not supported.

What is the maximum resolution of USB cameras that ESP32-S2/ESP32-S3 support?

- If we do not consider local JPEG decoding, the bottleneck is the throughput rate of USB. The USB camera generally adopts synchronous transmission. ESP USB has a limitation of FIFO size, which can reach 500 KB/s at the maximum currently. Thus, if you want to achieve 15 frames, the size of each frame can only be 33 KB. The maximum resolution that can be achieved by 33 KB depends on the compression rate, and generally it can reach 480 * 320.
- If you take local JPEG decoding into consideration, you also need to consider whether this resolution can reach 15 frames per second.

Can the ESP32-S2/ESP32-S3 USB recognize the USB plugging and unplugging action when it is used as a USB CDC Device?

- Yes, the USB device uses the tinyusb protocol stack, including mount and umount callback functions to response the USB plugging and unplugging events.
- Please note that if the device is a self-powered USB device and you need to detect the plug and unplug action without power off, please reserve the VBUS detection pin. Refer to the Self-Powered USB Device Solution.

After enabling the RNDIS and CDC functions on the ESP32-S3 USB, I found that the PC can recognize the COM port. However, the automatic programming function of the COM port is invalid. Is it expected?

- Yes. The USB auto-programming function is implemented through the USB-Seial-JTAG peripheral, and the USB RNDIS function is implemented through the USB-OTG peripheral. However, only one of the two peripherals can work at a moment.
- If the USB-OTG peripheral is used in the application, the automatic programming function implemented by the USB-Seial-JTAG peripheral will not be available. But you can manually enter the download mode for USB burning.
- Please refer to the USB-Serial-JTAG Peripheral Introduction.

Does ESP32-S2/ESP32-S3 support the USB CDC NCM protocol?

Please refer to the USB NCM example.

After I initialize the USB pins of ESP32-C3/ESP32-S3 to GPIO or other peripheral pins, why cannot I burn firmware through USB?

The USB pin of the ESP32-C3/ESP32-S3 can be initialized to GPIO or other peripheral pins. However, please note that after the initialization, the original USB download function will be disconnected, and the download mode cannot be entered automatically through USB. But you can manually pull down the Boot pin (GPIO9 in ESP32-C3 and GPIO0 in ESP32-S3) to make ESP32-C3/ESP32-S3 enter the download mode. Then you can download firmware through USB.

What should I pay attention to if I want to use the USB interface of ESP32-C3/ESP32-S3 as the unique download interface of firmware?

- DO NOT use USB pins of ESP32-C3 (GPIO18 and GPIO19) / ESP32-S3 (GPIO19 and GPIO20) as other peripheral functions.
- If the USB pins have to be reused as other functions in the application, the BOOT pin (GPIO9 in ESP32-C3, GPIO0 in ESP32-S3) must be wired to manually enter the chip into the download mode.

When I attempted to download and print log via the USB interface using the command `idf.py -p com35 flash monitor` on Windows, I encountered the following error. What's the reason for it?

- The error is as follows:

```
Connecting...
Failed to get PID of a device on com35, using standard reset sequence.
```

- On Windows, the COM port must be configured in the upper case, not the lower case `com`.

How can I apply for USB VID/PID for ESP32-S series products?

If your software is based on the TinyUSB protocol stack, you can use the default TinyUSB PID. Otherwise, you need to apply for a USB VID/PID for each ESP32-S series product. For detailed instructions, please refer to "usb-pids".

Is it possible to fix the COM port when downloading firmware using the USB-Serial-JTAG interface on Windows?

- Please open the Windows CMD as the administrator and execute the following command to add a registry entry. In this way, you can prevent incremental numbering based on the Serial number. Then you need to restart the computer to enable the modification.

```
REG ADD HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\usbflags\
↪303A10010101 /V IgnoreHWSerNum /t REG_BINARY /d 01
```

- For more information, please refer to Preventing Windows from incrementing COM numbers based on USB device serial numbers.

Can a USB drive be used for OTA upgrades?

Yes, it can be done using the component esp_msc_ota.

Does the ESP32 series chip support USB 2.0 High-Speed mode (480 Mbps)?

Currently, only ESP32-P4 supports USB 2.0 High-Speed mode.

How to improve the transmission rate of ESP32-S3 USB?

To enhance the transmission performance of USB, you can use the USB bulk transfer method, as well as increase the amount of data transferred per packet.

Does the USB interface of ESP32-S3 support USB charging function?

Not supported, currently the USB PD (USB-PowerDelivery) protocol is not supported.

A USB disk application is implemented based on the ESP32-S3 USB interface. Can the ESP32-S3 USB disk always appear as the Z: drive when plugged into a PC?

- The drive letter is automatically assigned by the operating system and cannot be fixed by the software on ESP32-S3.
- Drive Letter Assignment: The operating system will assign an unused drive letter to the USB disk based on the existing drive letters in the current system. In Windows systems, common drive letters are English letters from A to Z. Generally, A and B are traditionally reserved for floppy disk drives, C is usually the system disk, and the remaining letters are sequentially assigned to other storage devices (including hard disk partitions, optical drives, and external USB drives, etc.).

How does a USB UAC device synchronize with the host' s audio?

Since the USB bus is not a clock bus, the interval of each transmission is not fixed, so UAC devices may experience audio and video desynchronization, noise, and other phenomena. It is recommended to synchronize with the host using the feedback endpoint, and let the host send more or less data through the feedback endpoint to achieve audio synchronization.

The usb_device_uac component now supports the feedback endpoint. You can refer to the example code of this component to implement audio synchronization for the USB UAC device.

Does ESP32-P4 support USB?

Supported, ESP32-P4 has USB HS PHY and USB FS PHY as well as a USB-Serial-JTAG interface.

How to Use USB FS PHY as a Host with ESP32-P4?

When calling the USB host initialization function `usb_host_install`, you can use the USB FS PHY as the host by setting the value of `peripheral_map` in the structure parameter to `BIT1`.

Which two IOs are used when ESP32-P4 uses USB FS PHY as a host?

The default is IO26/IO27, which can be switched to IO24/IO25 by burning the efuse bit `USB_PHY_SEL`.

Can the two IO pins of the USB High-Speed PHY on ESP32-P4 be used as regular IO pins?

The USB_D- and USB_D+ of the high-speed USB2.0 OTG interface use dedicated digital pins, with pin numbers 49-50, and do not support use as regular IO ports.

Why can't the USB-Serial-JTAG function properly, or perform USB downloads and USB printing, after using USB-related features on the ESP32-S3?

Please refer to *usb_serial_jtag*.

Does the Espressif USB host protocol stack support HUB? How to use it?

Yes, the HUB function is supported in the low-level driver. To enable it, please set the macro `CONFIG_USB_HOST_HUBS_SUPPORTED=y`.

Can the Serial-JTAG and Full-Speed USB pins be adjusted on the ESP32-P4?

By default, `FS_PHY1` is connected to `USB_SERIAL_JTAG`, and `FS_PHY2` is connected to `OTG_FS`. The connection can be changed through `EFUSE_USB_PHY_SEL`.

Is the High-Speed USB of ESP32-P4 available for programming?

After entering download mode, programming can be done via High-Speed USB.

4.4.18 Other Peripherals

Can the REF_TICK clock frequency be modified?

CHIP: ESP32 | ESP32-S2 | ESP32-C3

No, the REF_TICK clock is fixed.

Does ESP32 support PCI-E protocol?

No, it doesn't.

Does ESP32-P4 support video encoding and decoding?

Yes. ESP32-P4 Supports JPEG hardware encoding/decoding, H.264 hardware encoding, and H.264 software decoding.

How to Use the Bit Manipulator on ESP32-P4?

The BitScrambler driver has been released. Please refer to BitScrambler Driver. The RMT peripheral now supports BitScrambler.

4.5 Protocols

4.5.1 ESP-TLS

When testing RTOS SDK `mqtt/ssl_mutual_auth` with ESP8266, the server connection failed. Why?

- The failure of SSL connection may due to insufficient memory of ESP8266.
- Please use the master version of ESP8266-RTOS-SDK to test this example, since it supports dynamic memory allocation in menuconfig so as to reduce the usage of memory peak. The specific action is: menuconfig -> `Component config` -> `mbedTLS` -> (type "Y" to enable) `Using dynamic TX /RX buffer` -> (type "Y" to enable) `Free SSL peer certificate after its usage` -> (type "Y" to enable) `Free certificate, key and DHM data after its usage`.

Can ESP HTTPS skip the server certificate check?

- Yes, if you enable the following options in menuconfig.
  - `Menu path: (Top)` -> `Component config` -> `ESP-TLS` -> `Allow potentially insecure options`
  - `Menu path: (Top)` -> `Component config` -> `ESP-TLS` -> `Allow potentially insecure options` -> `Skip server certificate verification by default`
- Besides, make sure that the `cert_pem` member variable is not set in the `esp_http_client_config_t` structure. Otherwise, the server certificate will still be verified with this CA certificate.
- If you want to test HTTP OTA at the same time, you need to enable the `Menu path: (Top)` -> `Component config` -> `ESP HTTPS OTA` -> `Allow HTTP for OTA` option in menuconfig.

How to set the `esp_tls_conn_read` API in ESP-TLS to non-blocking mode? Or is there any other way to implement non-blocking?

- You can set `non_block` to true in the `esp_tls_cfg_t` structure in `esp_tls.h` to achieve non-blocking.
- Alternatively, you can call `esp_transport_connect_async` to achieve non-blocking.

What are the TLS versions supported by ESP-IDF?

- The recommended TLS protocol in ESP-IDF is the Mbed TLS protocol.
- ESP-IDF v5.0 and later no longer support SSL 3.0, TLS 1.0 and TLS 1.1, but only support TLS 1.2 and TLS 1.3.

Why does the "mbedtls_ssl_handshake returned -0x7200" error occur during the SSL handshake?

- The 0x7200 error is caused by the following reasons:
  - MBEDTLS_SSL_IN_CONTENT_LEN is too small, generally less than the maximum 16 K
  - Insufficient memory
  - The server refuses the ESP connection and the sent SSL record is incomplete. You need to capture the packet and check it. For details, please refer to Espressif Systems Wireshark User Guide.

When accessing the MQTTS Broker using IP, the error `MBEDTLS_ERR_X509_CERT_VERIFY_FAILED` occurs, but communication can be successfully carried out under the same conditions in testing tools like MQTTX. What could be the reason?

The issue might be due to a mismatch in the CN field. This can be bypassed by setting the member variable skip_cert_common_name_check to True to skip the CN check.

How much memory is required for a stable TLS handshake on ESP32? How to solve the problem of insufficient memory during a TLS handshake?

A TLS handshake typically requires 40-50 KB of free heap memory for temporary allocation during the handshake process. If memory fragmentation is excessive or available memory is insufficient, the handshake may fail. When memory is insufficient, it needs to be optimized. Please refer to Minimizing RAM Usage. Additionally, if the module includes external PSRAM, ensure it is properly configured and utilized.

What do the error codes `0x6900` and `0x7600` represent during a TLS handshake?

- `0x6900 (MBEDTLS_ERR_SSL_WANT_READ)`: This usually indicates that the operation would be blocked because there is temporarily no data in the TCP receive buffer. It is not necessarily an error but rather a signal that more data needs to be received.
- `0x7600 (MBEDTLS_ERR_SSL_PRIVATE_KEY_REQUIRED)`: This indicates that the required private key or pre-shared key has not been set. For two-way authentication, please ensure that the client has configured the correct private key.

When implementing HTTPS on ESP32, how to resolve the issue that the server certificate chain is different from what is obtained by the PC browser?

Try enabling more detailed mbedtls debug information on ESP32 to obtain the certificate chain sent by the server during the TLS handshake process, and compare it with the certificate chain obtained from the browser.

How to resolve the issue of certificate parsing failure when using ESP32?

First, you should check whether the certificate format complies with the X509 standard and ensure that there are no issues with the certificate chain. In addition, enabling the log function of mbedtls can help diagnose the problem.

How to debug when TLS handshake and server certificate verification fail on ESP32?

Enable the Mbedtls debugging feature to log all activities during the TLS handshake, including the server certificate information received.

Does the MQTTS server require TLS authentication?

MQTTS servers usually require TLS authentication. If you want to skip the server CA certificate verification, you can set the CA certificate to NULL and enable `CONFIG_ESP_TLS_SKIP_SERVER_CERT_VERIFY` in menuconfig.

How to resolve the error `esp-tls: couldn't get hostname for drive.google.com: getaddrinfo() returns 202`?

This error indicates that DNS resolution has failed, which could be due to:

- The device's WiFi or LTE network is not correctly connected.
- lwIP only supports a single DNS server. When multiple network interfaces (such as WiFi + LTE) are used, the DNS configuration may be overwritten. It is recommended to check the DNS server configuration.

How can the ESP act as an HTTPS server and verify client certificates?

By default, when acting as a TLS server, the ESP does not validate the legitimacy of the client certificate. Even if client certificate verification fails, the ESP may still consider the handshake successful. To enable client certificate verification, you must provide a valid CA certificate via the `cacert_pem` configuration.

How to configure the HTTPS server's cipher suites to meet European certification requirements?

You can configure the supported cipher suites by calling `mbedtls_ssl_conf_ciphersuites()` after line 789 in esp_tls_mbedtls.c. Only enable high-security cipher suites such as `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`. Avoid using weak algorithms like SHA1 or MD5 to comply with security standards.

How can I detect if the peer has sent a FIN packet in a TLS connection?

You can check for the error code `-0x004C` or `errno=128`, which indicates that the peer has actively closed the connection.

How should I handle the expired TLS certificates after the device has been mass-produced?

- Extend the validity period of the certificate as much as possible to ensure that the certificate's validity period is longer than the device's service life;
- Update the certificate via OTA before it expires;
- If the certificate has expired, it can be updated through other remote methods, such as uploading a new certificate through the HTTP interface.

How to troubleshoot if a certificate error is prompted in a TLS or WSS connection, but the client configuration is confirmed to be correct?

You can use other TLS/WSS clients (such as a PC or third-party library) to connect to the same server for comparison testing. If the same certificate error is reported, it usually indicates an issue with the server's certificate or certificate chain, rather than a problem with the device's verification logic.

4.5.2 HTTP

Does ESP8266 support HTTP hosting?

Yes, it does. ESP8266 can run as a server in both SoftAP and Station modes.

- When running as a server in SoftAP mode, clients can directly access the ESP8266 host or server at 192.168.4.1 (default) IP address.
- When the server is accessed via a router, the IP address should be the one allocated to the ESP8266 by the router.
- When using SDK to write native code, please refer to relevant examples.
- When using AT commands, start a server using `AT+CIPSERVER` command.

How to use `esp_http_client` to send chunked data?

- Please use HTTP Stream by setting the `write_len` parameter of `esp_http_client_open()` to -1. Then the "Transfer-Encoding" will be set to "chunked" automatically, please see `http_client_prepare_first_line()` in esp_http_client.c.
- The code snippet is listed below for your reference:

```
static void http_post_chunked_data()
{
    esp_http_client_config_t config = {
    .url = "http://httpbin.org/post",
    .method = HTTP_METHOD_POST, // This is NOT required. write_len < 0 will␣
↪force POST anyway
```

(continues on next page)

(continued from previous page)

```
  };
  char buffer[MAX_HTTP_OUTPUT_BUFFER] = {0};
  esp_http_client_handle_t client = esp_http_client_init(&config);

  esp_err_t err = esp_http_client_open(client, -1); // write_len=-1 sets␣
↪header "Transfer-Encoding: chunked" and method to POST
  if (err != ESP_OK) {
    ESP_LOGE(TAG, "Failed to open HTTP connection: %s", esp_err_to_name(err));
    return;
  }

  // Post some data
  esp_http_client_write(client, "5", 1); // length
  esp_http_client_write(client, "\r\n", 2);
  esp_http_client_write(client, "Hello", 5); // data
  esp_http_client_write(client, "\r\n", 2);
  esp_http_client_write(client, "7", 1); // length
  esp_http_client_write(client, "\r\n", 2);
  esp_http_client_write(client, " World!", 7);  // data
  esp_http_client_write(client, "\r\n", 2);
  esp_http_client_write(client, "0", 1);  // end
  esp_http_client_write(client, "\r\n", 2);
  esp_http_client_write(client, "\r\n", 2);


  // After the POST is complete, you can examine the response as required␣
↪using:
  int content_length = esp_http_client_fetch_headers(client);
  ESP_LOGI(TAG, "content_length: %d, status_code: %d", content_length, esp_
↪http_client_get_status_code(client));

  int read_len = esp_http_client_read(client, buffer, 1024);
  ESP_LOGI(TAG, "receive %d data from server: %s", read_len, buffer);
  esp_http_client_close(client);
  esp_http_client_cleanup(client);
}
```

Is there a way to set cookies when ESP32 operates as an HTTP client?

ESP32 itself does not have an API for setting cookies directly, but you can use esp_http_client_set_header to add cookies to the HTTP header.

How do I set the maximum number of clients that are allowed for connection when ESP32 serves as an HTTP server? What will happen if the number exceeds the limit?

- The maximum number of client connections can be set by configuring `max_open_sockets` in the `httpd_config_t` structure.
- If the number of clients exceeds the limit, you can set the `lru_purge_enable` parameter in the `httpd_config_t` structure to true. In doing so, if there is no socket available (which is determined by max_open_sockets), the least used socket will be cleared to accept the coming one.

Does ESP32 have an example of implementing a gRPC client over HTTP/2 and above versions?

Not yet.

How to download a specific segment of a file over HTTP in ESP-IDF (i.e., add `Range:bytes` information to the header)?

Please refer to the `http_partial_download` function in the esp http client example.

When the ESP module acts as a local HTTP/HTTPS Server, it returns the *Header fields are too long for server to interpret* error if it is accessed by the browser. Why?

- The reason for this problem is that the URL is too long for the browser, and the underlying buffer is not big enough. You can increase the HTTP header length by modifying the menuconfig configuration. You can change the default value, which is 512 bytes, to a larger size, for example, 1024 bytes. The specific steps are as follows:
  - `idf.py menuconfig > Component config > HTTP Server > (1024) Max HTTP Request Header Length`
  - `idf.py menuconfig > Component config > HTTP Server > (1024) Max HTTP URI Length`

How can I resolve the error of *HTTP_HEADER: Buffer length is small to fit all the headers* returned by HTTP request?

- Please change the member `buffer_size_tx` of the structure `esp_http_client_config_t` in the file *esp-idf/components/esp_http_client/include/esp_http_client.h* to 1024 bytes or larger.

Why is the data length always 0 when calling `esp_http_client_read_response` after executing the `esp_http_client_perform` function?

The `esp_http_client_perform` function already handles data reading internally, so calling `esp_http_client_read_response` afterward is not necessary and will not yield the expected data length. To obtain the data, please handle the `HTTP_EVENT_ON_DATA` event in the event handler.

What happens when `esp_http_client_write` is called after calling `esp_http_client_open`?

Calling `esp_http_client_write` will immediately send data to the server. Typically, if `esp_http_client_perform` is called, the data will be passed to the application layer through `HTTP_EVENT_ON_DATA` in the HTTP event. If data is to be obtained using `esp_http_client_read`, it must go through `esp_http_client_open->esp_http_client_write->esp_http_client_fetch_headers`, after which``esp_http_client_read`` will contain data.

How to distinguish whether an HTTP request is sent from WiFi or from the Ethernet in the HTTP request handler?

In the request handler, you can get the file descriptor (fd) using `httpd_req_to_sockfd`, then call `getsockname` to get the local address. Based on the address, you can determine which network interface card is being used.

When downloading files via HTTP, occasionally there will be a failure with an error message "couldn't get hostname". What could be the reason?

This error is usually due to the failure of sending a DNS request, or WiFi not receiving a DNS response. It is recommended to confirm the problem through packet capture analysis.

In the event of an HTTP download failure, how to identify the cause?

First, analyze the logs to determine the cause of the download failure. If it is due to timeout or network disconnection issues, you can confirm whether the TCP stream is normal by capturing packets.

How to Upload a Grayscale Image via ESP HTTP?

The HTTP "Content-Type" can be modified as follows:

```
esp_http_client_set_header(http_client_cam, "Content-Type", "application/
↪octet-stream");
```

Alternatively, you can convert the image to JPEG format locally and continue using `image/jpeg` as the "Content-Type". If an `HTTP 500` error occurs during upload, verify that the image format is correct and ensure that the server supports this format.

How does an Espressif device handle HTTPS certificate expiration?

Certificates are usually updated via OTA. It is recommended to embed the certificates into app.bin, and then update app.bin through an OTA update to refresh the certificates.

How to resolve stack overflow in the HTTPD task?

To address this issue, increase the `stack_size` in `HTTPD_DEFAULT_CONFIG`. For details, refer to the example code.

What specification does ESP-IDF follow for URL encoding in HTTP/HTTPS requests?

The underlying URL encoding in ESP-IDF does not fully comply with the RFC-3986 specification for automatic encoding. Typically, users need to pre-encode the URL before passing it in.

If the status code of http_data is 200 but the content_length is 0, is this normal?

This situation may be normal, especially when the response uses chunked transfer encoding. In such cases, `content_length` may not be the accurate length. Use `esp_http_client_get_chunk_length` to obtain the actual length, or handle the data within the callback.

After running HTTPS for some time, a decrease in minimum heap memory is observed. Could this be a memory leak?

Not necessarily. If the remaining memory is stable, the behavior may simply indicate higher peak memory usage (e.g., TCP is in the TIME_WAIT state and not released). You can set the `SO_LINGER` option to force close the connection and release resources.

When downloading firmware via HTTP, a read timeout occurs, sometimes after reading about 5 KB of data (default TCP receive window 5760), and extending the timeout does not solve the problem. What could be the reason? How to solve it?

Possible reasons:

> Due to the working strategy of the mobile operator's base station, the Espressif device does not acknowledge the complete data packet with a full ACK. This causes intermediate network devices to drop the packet. Consequently, the server does not receive the ACK and stops transmitting subsequent data.

Troubleshooting Approach:

1. Switch carrier network for testing;
2. Check and increase the size of the TCP receive window (such as `CONFIG_LWIP_TCP_WND_DEFAULT`), and observe whether the amount of data read when failing increases with the window size.
3. Contact Espressif technical support to get guidance on the targeted solution.

How can HTTP connection reuse be maintained when performing a POST using low-level APIs like `esp_http_client_open`?

To reuse a connection, you can reuse the same `esp_http_client_handle_t` instance. As long as the server does not explicitly request the connection to close via the `Connection: close` header, and the ESP side does not call `esp_http_client_close()` function to close the connection, this connection will remain open for new requests. For more information, refer to Persistent Connections.

Do Espressif series chips support HTTP/2? Do they support WebSocket over HTTP/2?

Yes. You can use the espressif/sh2lib component. There is currently no WebSocket-over-HTTP/2 demo in the example, but the chip capability supports it, and users can implement it themselves. Regardless of the HTTP protocol version used, HTTP connections and WebSocket can coexist and listen on different ports respectively.

4.5.3 lwIP

How soon can the associated resources be released after the TCP connection is closed?

The associated resources can be released in 2 MSL, i.e. 120 seconds, or after the sent `linger/send_timeout` parameter is timeout.

After the SNTP calibration for ESP8266 RTOS SDK v3.2, errors gradually increase. How to resolve such issue?

This is because the ESP8266 uses software timer, which brings large errors itself. You can improve it with the following solutions:

- For branch v3.2, you can resynchronize time (300 s is recommended) from the server regularly by creating a task.
- For branch release-v3.3, the code of system timer has been refactored and is tested with low errors. On the other hand, you can still synchronize time from the server regularly.
- The master branch has inherited the refactored code from branch release-v3.3. In addition, you can configure the SNTP synchronization interval in menuconfig: `Component config` > `LWIP` > `SNTP` > `Request interval to update time (ms).`

Does ESP8266 support loop-back for the device end?

- Yes, it does.
- Please enable the LOOPBACK option from lwIP in menuconfig: `menuconfig` > `Component config` > `LWIP` > `Enable per-interface loopback` (type "Y" to enable).
- The device end sends data to the loopback address 127.0.0.1, and can read it from the address.

What is the default packet length for TCP/IP?

Please go to `menuconfig` > `Component config` > `LWIP` > `TCP` > `Maximum Segment Size (MSS)` for the length.

When using UTC and GMT methods in SNTP protocol, why can't I get the time of the target time zone ?

- The "TZ = UTC-8" refers to POSIX time, in which "UTC" is the abbreviation of any time zone and the number is the number of hours that the time zone is behind UTC.
- "UTC-8" indicates a certain time zone, "UTC" for short, which is -8 hours later than the actual UTC. Therefore, "UTC+8" is 8 hours later than the actual UTC, and also 16 hours later than Beijing.

Is there any special firmware or SDK in ESP32 that can only provide AP/STA (TCP/IP bypass) without using its internal TCP/IP so as to give developers more permissions?

The software solution ESP-Dongle can fit your requirements. Please contact Business Team to sign NDA and then get related solutions.

When ESP32 & ESP8266 are used as TCP servers, how can the ports be used again immediately after they are released?

- On both ESP32 and ESP8266, TCP ports are not immediately released after being closed. They remain in a TIME_WAIT state for a certain period of time. During this period, binding a socket with the same port and source address as before will fail. This is to ensure that you can receive the FIN signal sent by the server and can close the connection successfully. In this state, the port cannot be immediately reused. To address this issue, the socket option "SO_REUSEADDR" should be used, which allows the device to bind a TCP socket with the same port and source address in the TIME-WAIT state.
- Therefore, a TCP server program can set the "SO_REUSEADDR" socket option before calling bind() to bind the same port.
- Alternatively, the setsockopt() function can be used to set the SO_REUSEADDR option. Here is an example:

```
int reuse = 1;
if (setsockopt(socket, SOL_SOCKET, SO_REUSEADDR, &reuse, sizeof(reuse)) < 0) {
  ESP_LOGE(TAG, "setsockopt(SO_REUSEADDR) failed");
  return ESP_FAIL;
}
```

In the above code, "socket" means a socket that has already been created, and "reuse" is an integer variable with a value of 1, indicating that the SO_REUSEADDR option is enabled. If the setsockopt() function returns a negative value, it means that the setting has failed. Enabling the SO_REUSEADDR option allows the port to be immediately reused after being closed. However, there are some potential risks. If another connection uses the same port while it is still in the TIME_WAIT state, it may cause packet confusion. Thus, it's better to make choice based on actual situations.

After downloading the tcp_client example for an ESP32 module, I connected the module to the router via Wi-Fi and performed a Ping test on the computer. Then the it shows high latency sometimes, what is the reason?

When Wi-Fi is connected, Power Save mode will be turned on by default, which may cause high Ping delay. To solve this issue, you can turn off Power Save mode to reduce the delay by calling esp_wifi_set_ps(WIFI_PS_NONE) after `esp_wifi_start()`.

How can I set static IP when using ESP-IDF?

For details, please refer to static_ip example.

Does ESP32 have an LTE connection demo?

- Yes. For ESP-IDF v4.2 and later versions, please refer to the example/protocols/pppos_client demo.
- For ESP-IDF v5.0 and later versions, please refer to examples in the *esp-protocols* repo.

Will memory leak occur when ESP32 TCP repeatedly closes and rebuilds socket (IDF 3.3)?

In ESP-IDF v3.3, every time a socket is created, a lock will be assigned, given that this internal socket array has not been assigned any lock before. This lock will not be reclaimed after the socket is released. Thus, next time the same socket array is allocated, the previous lock will be used again. That is to say, every time a new socket array is allocated and released, there will be one lock memory used. After all socket arrays being allocated, there will be no memory leak any more.

Are there any limits on the maximum number of TCP client connection after the ESP32 additionally opens the TCP server?

Yes. The number of simultaneously connected socket fd number for ESP32 is limited by `LWIP_MAX_SOCKETS`, which is 10 by default.

What is the default MTU of lwIP for an ESP32?

The default MTU of lwIP is 1500. This is a fixed value and it is not recommended to change it.

How to increase the DNS request time for ESP32?

You can manually modify the `#define DNS_MAX_RETRIES 4` in esp-idf/components/lwip/lwip/src/include/lwip/opt.h. For example, you can change the value of `#define DNS_MAX_RETRIES` to 10. In this way, the maximum time that DNS waits for a response from the server is 46 s (1+1+2+3+4+5+6+7+8+9).

After creating and closing TCP SOCKET several times, an error is reported as "Unable to create TCP socket: errno 23" . How to resolve such issue?

:CHIP: ESP8266 | ESP32 | ESP32-S2 | ESP32-C3 | ESP32-S3 :

- Reason: "errno 23 " means open many open files in system. Closing a socket takes 2 MSL of time, which means sockets will not be closed immediately after calling the close interface. Due to this reason, open sockets are accumulated and exceeds the maximum connection number (the default is 10 in menuconfig, the maximum connection is 16) thus triggering this error.
- Solution: Set SO_LINGER via the setsockopt interface to adjust the TCP close time.

```
linger link ;
link.on_off = 1 ;
link.linger = 0 ;
setsockopt(m_sockConnect, SOL_SOCKET, SO_LINGER, (const char*)&link,␣
↪sizeof(linger));
```

What happens when ESP8266 receives a "tcp out of order" message?

- If `CONFIG_LWIP_TCP_QUEUE_OOSEQ(Component config -> LWIP -> TCP -> Queue incoming out-of-order segments)` is enabled, the out-of-order messages will be stored at the cost of memory consumption.
- If this configuration is disabled, after receiving the "out of order" message, data will be discarded and a retransmission will be requested. For example, there are four data packets namely 1, 2, 3 and 4, ESP8266 receives 1 first, and then receives 4. If this configuration is enabled, ESP8266 will store the data of 4, wait until it receives 2, 3, and then report the four packets to the application layer; if this configuration is disabled, ESP8266 will discard the packet of 4 when it receives it, and let the other side send packet 2, and then the other side will send from 2. Under this condition, the retransmission is increased.

Does ES32 support PPP functionality?

Yes, please refer to esp_modem example.

Every time ESP32 attempts to read 4 KB of data with `read` and `recv` APIs in the socket, it can not always read the 4 KB of data. Why?

- Both `read` and `recv` APIs are used to read the data in the underlying buffer. For example, if there are 100 bytes of data in the underlying buffer, and the `len` size passed in by `read` and `recv` is only 50, then the API will return after it reads 50 bytes. If `len` exceeds the length of the data received in the underlying buffer, say 200, the API will return after it reads 100 bytes and will not wait until it receives 200 bytes. So, an attempt to read 4 KB of data will not necessarily return 4 KB of data, but only the data available in the underlying buffer at the time of reading.
- If you need to read 4 KB of data every time, it is recommended to use application code on top of the socket layer to design the corresponding logic, which reads data recursively until it reaches 4 KB.

What is the version of lwIP currently used in ESP-IDF?

The current version of lwIP in use is 2.2.0.

In DHCP mode, will ESP32 renew the IP or apply for a new IP when the lease expires?

There are two lease periods, T1 (1/2 time of the lease) and T2 (7/8 time of the lease) in DHCP mode. When both lease expires, ESP32 usually renews the same IP. Only when both of them fail to renew will ESP32 apply for a new IP.

Why does ESP-IDF report an error when `SO_SNDBUF` option of `setsockopt` are used to get or set the size of the send buffer?

By default, lwIP does not support `SO_SNDBUF`. To set the send buffer size, go to `menuconfig` -> `Component config` -> `LWIP` -> `TCP` -> `Default send buffer size`. To get or set the receive buffer size, you need to enable the `CONFIG_LWIP_SO_RCVBUF` option in menuconfig before you can use the `SO_RCVBUF` option of `setsockopt` to get or set the receive buffer size.

I find that the network data latency of TCP & UDP is large when testing ESP-IDF. What is the buffering data mechanism of TCP & UDP protocols?

- For TCP, there is `TCP_NODELAY` option in socket option. You can enable this option to disable the Nagle algorithm which is enabled by default, so that the data will not be cached locally and then sent together.
- For UDP, UDP data is sent directly. If there is any delay, it is because of the delay of the Wi-Fi network environment, not the UDP itself.
- If TCP retransmission is caused by poor network environment, and the transmission interval is set too long, high latency will occur. You can try to shorten the RTO value (by modifying `component config` -> `lwip` -> `tcp` -> `Default TCP rto time` and `TCP timer interval` options in menuconfig).

How to choose the default route when ESP32 works as dual NICs (e.g. ETH+STA)?

The following summarizes how the default route is selected for dual NICs, using ETH and STA as examples.

- Supposing ETH and STA are in the same LAN:
  - When the device accesses the LAN address, the data will go to the last up netif.
  - When the device accesses the non-LAN address, the data will go to the netif with the larger `route_prio` value.
- Supposing ETH and STA are not in a LAN, ETH belongs to 192.168.3.x segment, and STA belongs to 192.168.2.x segment:
  - When the device accesses 192.168.3.5, it will take the ETH netif.
  - When the device accesses 192.168.2.5, it will take the STA netif.
  - When the device accesses 10.10.10.10, it takes the default route (the netif with the larger `route_prio` value). When netif is up, it sets the default route based on the `route_prio` value size, and the default route is often the netif with the larger `route_prio` value. When the device accesses an address that is not inside the routing table, the data takes the default route.

How do I enable keepalive for TCP in ESP-IDF?

You can refer to the code for enabling TCP keepalive in esp_tls.c.

Is it possible to operate the same socket in multiple threads in ESP-IDF?

In ESP-IDF, it is possible for multiple threads to share one single socket for communication. Each thread can use the same socket to send and receive data, but it is important to ensure thread synchronization when accessing the socket to avoid race conditions and deadlocks. Typically, a mutex can be used to control access to the socket, ensuring that each thread's access to the socket is mutually exclusive to avoid data corruption caused by concurrent access to the socket. However, operating on the same socket from multiple threads is risky, and it is not recommended.

How much time do ESP devices allocate to other device's IPs in ESP DHCP server mode?

The default is 120 s. Please refer to the `DHCPS_LEASE_TIME_DEF` parameter, which is not recommended to be set to a small value.

What are the three lease related times in ESP-IDF DHCP? What parameters in the code do they correspond to?

They are Address Lease Time, Lease Renewal Time and Lease Rebinding Time, corresponding to the lwIP codes `offered_t0_lease`, `offered_t1_renew`, and `offered_t2_rebind` respectively.

What is the maximum length for each data transmission in the ESP-IDF lwIP?

If you are using the socket interface `send`, the maximum length supported is determined by the `SSIZE_MAX` parameter. If you use the `tcp_write` function, the maximum length is limited by `snd_buf` (send buffer length). `send` is a socket interface wrapped by lwIP based on the sequential API, which is a higher-level interface than `tcp_write` and is more suitable for user-level calls. There is basically no difference in resource usage between the two API calls.

If I need more debug logs for lwIP layer related issues with ESP-IDF, how can I enable the corresponding debug log to be printed (e.g. DHCP, IP)?

- To print lwIP-related debug log, open menuconfig, go to `Component config` -> `LWIP`, and enable the option `Enable LWIP Debug`. There are sub-options, including `Enable IP debug messages` and `Enable DHCP debug messages`. You could enable them as needed.
- If you don't find the desired debug log module in the above menuconfig, such as, UDP module, first check if there is `#define UDP_DEBUG` in `esp-idf/components/lwip/port/esp32/include/lwipopts.h`. If yes, change `#define UDP_DEBUG LWIP_DBG_OFF` to `#define UDP_DEBUG LWIP_DBG_ON` manually. If no, add `#define UDP_DEBUG LWIP_DBG_ON` to `esp-idf/components/lwip/port/esp32/include/lwipopts.h` referring to `#define UDP_DEBUG LWIP_DBG_OFF` in esp-idf/components/lwip/lwip/src/include/lwip/opt.h file.

What is the difference between socket blocking and non-blocking in ESP-IDF?

- For reads, the difference is whether the read interface returns immediately when no data arrives at the bottom. A blocking read will wait until data has arrived or until an exception occurs, while a non-blocking read will return immediately with or without data.
- For writes, the difference is whether the write interface returns immediately when the underlying buffer is full. For blocking write, if the underlying buffer is not writable (the underlying buffer is full or the peer has not acknowledged the previously sent data), the write operation will keep blocking until it is writable or an exception occurs. For non-blocking write, it will write as much as it can without waiting for the underlying buffer to be writable or the length of the write to be returned.
- The non-blocking interface call does not block the current process, while the blocking interface does.

Can ESP32 use the IP of the previous successful connection for communication after connecting to the router, and in case of failure, re-enter the authentication process and use DHCP to obtain a new IP?

- Yes, if you enable `Component config` -> `LWIP ->DHCP: Restore last IP obtained from DHCP server` option in menuconfig.
- Note that you cannot use a static IP instead, because static IP settings do not have conflict detection. It may lead to IP conflict.

How do I achieve connect_timeout when programming with sockets?

- If you set the socket to the non-blocking mode, the connect() function will also be non-blocking. Then you can set the timeout by the select() function to determine whether the socket is connected successfully or not. For details, please refer to "connect_timeout settings of sockets" .

When ESP32 uses SNTP to synchronize the current time, I found that there is a random delay. After further analysis, I found that it is caused by `SNTP_STARTUP_DELAY` in the IDF lwip component, the default value of which is 1. Is there any way to avoid the random delay without modifying the IDF component?

- There is no way to avoid the random delay without modifying the IDF component. You need to manually add the code `#define SNTP_STARTUP_DELAY 0` to lwipopts.h in the lwip component. This code reduces the time that SNTP takes to send a request, so it can reduce the total time for ESP devices connecting to the cloud after they are powered up as a result.
- The reason for enabling this random delay option by default is that it is mandated by the SNTP RFC protocol. A random delay can reduce the number of simultaneous accessing devices, so this can prevent the SNTP server from being overloaded.

Do IPv4 and IPv6 support setting a static IP?

- If it is a local static IP, IPv4 supports manual configuration, but the local IP for IPv6 is automatically generated according to protocol rules and does not require manual configuration.
- If it is a global static IP, both IPv6 and IPv4 support manual configuration.

After connecting an Android phone to ESP SoftAP, the phone will prompt "No Internet access, do you want to continue using it?" If I choose "Do not use" , the phone can still access external web pages through cellular data. However, I cannot access external web pages through an Apple phone with the same operation. What is the reason for this? How to solve it?

- Reason: The DHCP ACK returned by the ESP DHCP Server includes the option3 (router) field. Therefore, after the mobile phone parses this option, it will set the default route to 192.168.4.1, causing the Apple mobile phone to access external web pages through ESP Wi-Fi instead of cellular data.
- Solution: You can comment out the dhcpserver.c code snippet.

TCP or UDP transmission fails with the error code 12(ENOMEM). How to solve it?

12 indicates insufficient memory. First, print the remaining internal memory. If the memory is sufficient, the error is caused by a full Wi-Fi TX buffer. In this case, please send data on the application layer more slowly or increase the Wi-Fi TX buffer in sdkconfig.

What usually causes the *transport_base: Poll timeout or error, errno=Connection already in progress* error? How to determine if a poll timeout is due to network issues or code logic problems?

This error is typically caused by an unstable network connection or a full underlying buffer. A poll timeout indicates that no ACK was received from the peer or the peer did not respond while waiting for the write operation to complete. Packet capture analysis can be used to determine whether the ACK from the peer was not received or the peer did not respond. If it's a network issue, you can optimize the network environment or add a retry mechanism; if it's a code logic issue, you need to check whether the use of poll and select in the code is correct, ensuring that the timeout period and retry strategy are reasonable.

From which serial number does the ESP32 LWIP SOCKET start?

By default, the socket number starts from 54. The macro that determines the socket number is `LWIP_SOCKET_OFFSET`, which can be calculated by subtracting CONFIG_LWIP_MAX_SOCKETS from FD_SETSIZE. In the `include/sys/select.h` file of the compilation toolchain, the value of `FD_SETSIZE` is 64, and the default value of `CONFIG_LWIP_MAX_SOCKETS` is 10. Therefore, by default, the value of `LWIP_SOCKET_OFFSET` is 54. When `CONFIG_LWIP_MAX_SOCKETS` is set to 16, the value of `LWIP_SOCKET_OFFSET` becomes 48, and the socket number starts from 48.

How to optimize the TCP send buffer?

Try the following configurations:

- CONFIG_ESP_WIFI_STATIC_RX_BUFFER_NUM = 16
- CONFIG_ESP_WIFI_DYNAMIC_RX_BUFFER_NUM = 64
- CONFIG_ESP_WIFI_DYNAMIC_TX_BUFFER_NUM = 64
- CONFIG_ESP_WIFI_AMPDU_TX_ENABLED = y
- CONFIG_ESP_WIFI_TX_BA_WIN = 32
- CONFIG_ESP_WIFI_AMPDU_RX_ENABLED = y
- CONFIG_ESP_WIFI_RX_BA_WIN = 32
- CONFIG_LWIP_TCP_SND_BUF_DEFAULT = 28800
- CONFIG_LWIP_TCP_WND_DEFAULT = 28800
- CONFIG_LWIP_TCP_RECVMBOX_SIZE = 32
- CONFIG_LWIP_TCPIP_RECVMBOX_SIZE = 32

When ESP32 is used as an STA, Could a static IP trigger the GOT_IP event?

If the static IP does not conflict with other devices on the network, a GOT_IP event can be triggered; if there is a conflict, the GOP_IP event cannot be triggered.

What could cause a TCP write operation timeout?

A TCP write operation timeout is typically caused by network latency or the peer device failing to acknowledge (ACK) in time. If the timeout is set too short (e.g., 200 ms), the write operation may fail. It is recommended to increase the timeout duration and check network conditions to ensure that the peer device responds in time.

What could cause a TCP write operation returning -1?

Returning -1 usually indicates that the socket has been closed. This may happen due to a previous write operation timeout, or a network issue causing the socket to close. It is recommended to check the network connection and ensure that the socket is valid before use.

Why does the WebSocket callback receive packets of varying sizes, sometimes split, sometimes combined?

This is a normal behavior of TCP transmission. WebSocket operates over TCP, which transmits data as a stream. Therefore:

- The received packet size can vary because TCP may split or merge packets during transmission.
- The WebSocket framework does not control this behavior. If a fixed packet size is required, you need to implement buffering and packet reassembly at the application layer yourself.

Does ESP32 support using different DNS servers for each network interface?

Yes. Starting from ESP-IDF v5.2, the `esp_netif` layer stores the DNS server addresses obtained by each network interface. However, in the Lwip layer, all network interfaces still share a single DNS server. The `esp_netif` layer sets the DNS server of the Lwip layer based on the default route interface (i.e., the interface with the highest priority). To enable this feature, please define the macro `CONFIG_ESP_NETIF_SET_DNS_PER_DEFAULT_NETIF`.

When ETH, Wi-Fi, and PPP connections are all successfully established in the system, which network interface will be used for application-layer data, such as MQTT? Is there a priority mechanism? How can it be configured?

Please refer to: Analysis of Default Route Selection Mechanism for Dual Network Interfaces on ESP32

When DNS resolution fails, what aspects should be checked to identify the cause?

Packet capture analysis is one of the key steps. At the same time, check whether there is enough memory allocated for related operations. For instance, a failure in pbuf_alloc allocation may be due to insufficient memory.

Why does the device fail to respond to Ping when AP + STA are in the same network segment?

This is usually due to a routing conflict. To avoid this, ensure that the AP and STA are assigned to different subnets. If using the same subnet is required, you must explicitly bind the outgoing interface at the application layer—note that this is not supported by standard lwIP implementations.

Why does the device open multiple UDP ports? Is this normal?

This is normal behavior. UDP ports are used by protocols such as DHCP, DNS, and SNTP. For each request, a temporary port (typically above 1024) may be allocated, which will automatically close after the request is completed.

How should I handle the situation where the ESP32-P4 cannot obtain a global IPv6 address when using an Ethernet connection, but the link-local IPv6 address is available?

You can check whether line 1176 of the esp_netif_lwip.c file is executed in the Ethernet scenario. At the same time, confirm the value of `ip6_autoconfig_enabled` (0 or 1).

When there are multiple netifs (such as Wi-Fi and Ethernet) on a device, can the data be automatically switched to another netif when one netif is disconnected?

Automatic switching is not supported. The direction of data flow is determined by the priority of netif. If the high-priority netif remains connected (even if it is actually offline), data will still attempt to go through that interface, and will not automatically switch to another available lower-priority interface.

When both Wi-Fi and Ethernet are enabled, and a static IP is set for Ethernet, DNS resolution fails. How should this be troubleshooted and resolved?

In multi-netif scenarios, the DNS server obtained by the netif started later may overwrite the previous one. The solution is as follows:

1. Check and enable the `LWIP_DNS_SETSERVER_WITH_NETIF` configuration, so that each netif has its own independent DNS server.
2. Ensure that the code for setting a static IP is executed after the correct network event (such as `ETHERNET_EVENT_CONNECTED`).

How can I publish the mDNS service on a specific network interface (netif)? For example, how to have different services published on Wi-Fi or Ethernet interfaces respectively?

The current mDNS implementation does not support registering different services on different netifs. Services are registered globally, and only the enabling or disabling of mDNS on a netif can be controlled (through `mdns_netif_action`). It is not possible to have "Service A only on Wi-Fi, and Service B only on Ethernet".

How to update headers (such as Authorization token) before automatic reconnection when using esp_websocket_client?

The newer version of the `esp_websocket_client` component provides the `esp_websocket_client_set_headers` API, which allows for dynamic updating of header information when needed, without the need to destroy and rebuild the client.

Do ESP32 series chips support scenarios such as obtaining global IPv6 address prefixes and DNS server addresses in IPv6 mode?

The acquisition of global IPv6 address prefixes only supports the SLAAC method, while the acquisition of DNS server addresses can support both SLAAC and DHCPv6 stateless methods.

Does ESP-IDF support the SNMP protocol stack?

lwIP has built-in SNMPv3 support, but it is not enabled at the ESP-IDF level and no example is provided. Users need to call the SNMP interfaces in lwIP themselves.

When receiving UDP packets over Ethernet, out-of-order delivery (with packet loss) is observed. How to determine whether the packet loss occurs at the lwIP layer or the Ethernet layer?

Reduce the `CONFIG_LWIP_TCPIP_RECVMBOX_SIZE` and `CONFIG_LWIP_UDP_RECVMBOX_SIZE` (for example, from 48 to 16). If the received packets also decrease accordingly, it indicates that packets are being dropped at the lwIP layer; otherwise, the problem is at the Ethernet layer.

4.5.4 Mbed TLS

Does ESP8266 OpenSSL support hostname validation?

Yes. ESP8266 OpenSSL is based on Mbed TLS encapsulation, which supports `hostname validation`. ESP-TLS can be used to switch between Mbed TLS and wolfSSL.

How to optimize memory when ESP32 uses Mbed TLS?

- You can enable dynamic buffer in menuconfig, the specific operation is `menuconfig > Component config > mbedTLS > Using dynamic TX/RX buffer (key "Y" to enable).`
- At the same time, you can enable the sub-options `Free SSL peer certificate after its usage` and `Free certificate, key and DHM data after its usage` in the `Using dynamic TX/RX buffer` in the previous step.
- However, ESP-IDF v5.0 and later no longer have sub-optioin `Free SSL peer certificate after its usage`, and Mbed TLS enables `MBEDTLS_SSL_KEEP_PEER_CERTIFICATE` by default. If you want to save memory, you can close it by `menuconfig > Component config > mbedTLS > mbedTLS v3.x related > Keep peer certificate after handshake completion (key "N" to disable)` .

When I connected an ESP32 module with the HTTPS Server, I got the following log. What is the reason?

```
free heap size: 181784 bytes
I (4285) esp_https_server: Starting server
E (4285) esp_https_server: Could not allocate memory
I (4295) example: Error starting server!
I (4295) SSDP Server: SSDP server started
free heap size: 178636 bytes
```

- The error is caused by low memory. The log shows that you use the esp_get_free_heap_size() API to get the remaining memory. However, the remaining memory includes the chip's internal RAM as well as external PSRAM.
- By default, mbedTLS uses internal RAM memory, and you can use the esp_get_free_internal_heap_size() API to obtain the remaining internal memory.
- If the module has an external PSRAM, you can modify the configuration from `menuconfig > Component config > mbedTLS > Memory allocation strategy > Internal memory` to `menuconfig > Component config > mbedTLS > Memory allocation strategy > External SPIRAM` for testing.

When resolving a hostname on ESP32, I encountered the following error. What could be the reason?

```
getaddrinfo() returns 202, addrinfo=0x0
```

- The error is caused by DNS request timeout.
- You can enable DNS log with the debug level or capture wireless packets for further analysis.
- To enable the debug level DNS log, you can add `#define DNS_DEBUG LWIP_DBG_ON` code to the `esp-idf/components/lwip/lwip/src/include/lwip/opt.h` file, and then enable the `Component config > LWIP > Enable LWIP Debug` configuration.

Why does the following mbedtls software error occur when I develop applications based on the esp-idf SDK?

```
E: esp-tls-mbedtls: mbedtls_ssl_handshake returned -0x4290
E: esp-tls: Failed to open new connection
E: transport_base: Failed to open a new connection
E: HTTP_CLIENT: Connection failed, sock < 0
E: HTTP_CLIENT: Failed to open HTTP connection: ESP_ERR_HTTP_
↪CONNECT
```

- The mbedtls error code is 0x4290, which is generally 0x4280 + 0x10. 0x4280 represents the error stage MBEDTLS_ERR_RSA_PUBLIC_FAILED, and 0x10 indicates that the cause of this error stage is MBEDTLS_ERR_MPI_ALLOC_FAILED, which means that this error stage is caused by a failure in memory allocation.
- For the meanings for mbedtls error codes, refer to Mbed TLS error codes.

The following error occurred when I ran the esp-idf/examples/protocols/https_mbedtls example on ESP32 to connect to a TLS v1.3 server in the ESP-IDF v5.1.2 SDK. What could be the reason?

```
E(53769) example: mbedtls_ssl_handshake returned -0x6c00
I(53779) mbedtls: ssl_tls.c:355 Reallocating in_buf to 4429
I(53779) mbedtls: ssl_tls.c:355 Reallocating in_buf to 16717
E(53769) example: Last error was: -0x6c00 - SSL - Internal error␣
↪(eg, unexpected failure in lower-level module)
```

- TLS v1.3 is not yet supported on ESP-IDF v5.1.2. If you need to connect to a TLS v1.3 server, please test with the ESP-IDF v5.2-beta1 or later SDK. See: esp_tls: add initial support for TLS 1.3 connection.

How can mbedtls be modified to support the parsing of fragmented TLS handshake messages?

You can refer to the relevant PR from the mbedtls community (e.g., https://github.com/Mbed-TLS/mbedtls/pull/9872/files). This PR provides a solution.

Does ESP32 support the secp256r1 encryption algorithm?

Yes, ESP32 supports the secp256r1 encryption algorithm, and the corresponding implementation is provided in mbedtls.

What does the error `mbedtls_ssl_handshake returned -0x2800` indicate?

This error is usually caused by insufficient memory. It is recommended to check the available RAM on the device.

How to restrict mbedtls to only use specific cipher suites?

You can use the *mbedtls_ssl_conf_ciphersuites* interface, as shown below:

```
static int force_ciphersuite[2] = {MBEDTLS_TLS_RSA_WITH_AES_128_GCM_
↪SHA256, 0};
mbedtls_ssl_conf_ciphersuites(&tls->conf, force_ciphersuite);
```

For more details, please refer to: https://github.com/espressif/esp-idf/blob/master/components/esp-tls/esp_tls_mbedtls.c#L880-L889.

Using a certificate validated by OpenSSL, the device fails to verify the certificate when connecting to the MQTT server. However, the MQTTX tool can successfully connect and verify it. Why?

Check whether the field length of the SAN (Subject Alternative Name) IP address in the certificate meets the definition of the x.509 standard. In Mbedtls, the length for IPv4 is 4 bytes, and the length for IPv6 is 16 bytes.

Is the WolfSSL license included with Espressif chips?

The agreement between Espressif chips and WolfSSL has expired. You need to purchase their license for commercial usage.

During OTA upgrade, Mbedtls reports an error (e.g., -0x7780). How to resolve this?

0x7780 indicates that the device received an alert message sent by the peer. For example, when the Espressif device uses a self-signed CA certificate and acts as an HTTP server, if a PC browser accesses the server, the PC will send such an alert message. However, this does not affect normal communication. In other cases, it is recommended to enable Mbedtls debug logs to investigate further.

TLS handshake failed with error code -0x7100 (BAD_INPUT_DATA). What could be the reason?

This is usually caused by incomplete reception of data, making parsing impossible. Verify whether `CONFIG_MBEDTLS_SSL_IN_CONTENT_LEN` is set to 16 K. If it is already configured to the maximum 16 K, enable Mbedtls debug logs to investigate the issue.

The certificate works fine when connecting to a local MQTT broker via MQTTX, but on the ESP device, it reports `mbedtls_ssl_handshake returned -0x7A00`. Is this a certificate format issue?

The error code `-0x7A00` indicates a certificate validation failure, which is not necessarily a format issue. Troubleshooting suggestions: 1. Enable mbedTLS log. 2. Enable the debug log in the `esp_mbedtls_verify_certificate` function in ESP-IDF to obtain detailed failure reasons.

4.5.5 MQTT

How to configure the server address so as to make it an autonomic cloud platform by using MQTT?

Please refer to MQTT Examples.

I'm using ESP8266 release/v3.3 version of SDK to test the example/protocols/esp-mqtt/tcp example. Then during Wi-Fi configuration, the connection fails after configuring SSID, password and connecting to the default server. The log is as follows, what is the reason?

```
W (4211) MQTT_CLIENT: Connection refused, not authorized
I (4217) MQTT_CLIENT: Error MQTT Connected
I (4222) MQTT_CLIENT: Reconnect after 10000 ms
I (4228) MQTT_EXAMPLE: MQTT_EVENT_DISCONNECTED
I (19361) MQTT_CLIENT: Sending MQTT CONNECT message, type: 1, id: 0000
```

When such error occurs, it indicates that the server rejected the connection because the client's wrong MQTT username and password caused the server-side authentication to fail. Please check if you are using the correct MQTT username and password.

What is the default keepalive value of the MQTT component in ESP-IDF?

The default value is 120 s, defined by the `MQTT_KEEPALIVE_TICK` macro in `mqtt_config.h`. This value can be modified via the `session.keepalive` parameter in `esp_mqtt_client_config_t`.

Does MQTT support automatic reconnection?

- Yes. The automatic reconnection of MQTT is controlled by the member variable `disable_auto_reconnect` in esp_mqtt_client_config_t. The default value of this variable is `false`, which means automatic reconnection is enabled.
- You can also modify the reconnection timeout by changing the `network.reconnect_timeout_ms` parameter in `esp_mqtt_client_config_t`.

What are the supported MQTT versions of ESP-IDF?

ESP-IDF currently supports MQTT 3.1 and MQTT 3.1.1, and MQTT 5.0.

When a Wi-Fi connection is disconnected in ESP-IDF, will the memory previously requested by MQTT upper layer protocol be automatically released?

- No, but you do not need to care about this memory. What you need to care about is the application layer that ESP encapsulates.
- For MQTT application layer components, you get an MQTT handle when initializing MQTT. You only need to care about the memory in this handle. When not using MQTT, you can call `stop` or `destroy` to release the corresponding MQTT memory. When Wi-Fi is disconnected and connected, you do not need to release the MQTT memory or reapply for the handle, because there is an automatic reconnection mechanism in the MQTT component.

For ESP32-C3 MQTT, can I not set corresponding `client_id` but configure it as an empty string by default?

Yes. You can achieve this by setting the `credentials.set_null_client_id` parameter of `esp_mqtt_client_config_t` to `true` in the application code.

When `MQTT_EVENT_PUBLISHED` is triggered after an ESP-IDF MQTT client has published data with QoS of 1 or 2, does it mean that a proper ack has been received from the other side to prove that the publish has completed? Or does it just mean that the data was successfully sent to the server once?

The `MQTT_EVENT_PUBLISHED` event triggered means that the broker has acknowledged receipt of the messages published by the client, proving that the publish has completed successfully.

How does an ESP MQTT client manually release MQTT resources after disconnection?

Calling the esp_mqtt_client_destroy API will do the trick.

How should I configure the MQTT keepalive time when ESP32 Wi-Fi and Bluetooth LE coexist? Is there any appropriate configuration time?

- When using Wi-Fi and Bluetooth LE concurrently in ESP32, it is recommended to configure the MQTT keepalive time properly. Since both Wi-Fi and Bluetooth LE require system resources, setting the keepalive time too short may cause high system load, affecting system stability and performance.
- Generally, it is advisable to set the MQTT keepalive time based on actual needs to ensure the device stays online while minimizing system resource consumption. In the case of Wi-Fi and Bluetooth LE coexistence, it is recommended to set the MQTT keepalive time to a longer duration, such as 30 seconds or 60 seconds, to reduce communication between the device and the MQTT broker, thereby reducing system load.
- It is important to note that setting the keepalive time too long may cause a delay in detecting the device offline when it disconnects, which may affect real-time performance and reliability. Therefore, the MQTT keepalive time should be set based on actual needs and system performance.

When will the disconnect event message be triggered for ESP-MQTT clients?

The disconnect message only occurs in the follow cases:

- A TCP connection error occurs while the MQTT connection is being established.
- An MQTT connection error occurs while the MQTT connection is being established.
- You actively call the `disconnect` function.
- An exception is received or sent.
- The MQTT `PING RESPONSE` is not received within the specified time.
- The MQTT `PING` request failed to be sent.
- Reconnection.

Does the ESP32 MQTT client automatically try to reconnect after disconnecting from the server?

Yes. The ESP MQTT client controls whether to enable automatic reconnection by the `network.disable_auto_reconnect` parameter in the `esp_mqtt_client_config_t` structure. By default, this parameter is set to `false`, indicating that MQTT will automatically attempt to reconnect.

How to check if the ESP32 is disconnected from the MQTT server?

You can use the MQTT `PING` mechanism for detection. Specifically, in the `esp_mqtt_client_config_t` structure, set `session.disable_keepalive` to `false` (default value, which means the keepalive mechanism is enabled), and configure the `session.keepalive` parameter to 120 s (default value). In this way, the MQTT client will periodically send `PING` messages to check the connection status with the server.

How can I identify the specific error and troubleshoot the issue when encountering a connection failure to the MQTT server?

When MQTT connection fails, you can parse the data in the `esp_mqtt_error_codes_t` structure within the MQTT event `MQTT_EVENT_ERROR`. For more details, please refer to the `MQTT_EVENT_ERROR` event in the example.

How to adjust the sending time of MQTT' s will message?

The delay in sending will messages can be reduced by shortening the MQTT heartbeat time.

While sending data, the MQTT client encounters a timeout. How to determine which network layer the problem occurred in?

Issues can be identified by packet capture analysis to determine whether they occur at the transport layer, network layer, or data link layer. The specific failure point could be that the server did not return an ACK, the server returned an ACK but Wi-Fi did not receive it, or the data packet was not successfully sent.

Why does the MQTT client still report a write timeout when the network condition is good?

This may be due to the underlying LWIP buffer being full, resulting in an inability to write. This is usually because the packets in the buffer have not received an ACK from the other end. Specific reasons may include the server not sending an ACK, the server sending an ACK but Wi-Fi not receiving it, or the data packet not being successfully sent out.

Why does memory usage drop sharply when using MQTT communication?

The outbox in MQTT will occupy memory. For messages with QoS greater than 0, the related memory will only be released after the ACK from the MQTT layer is sent by the peer. You can track the allocation and release of memory by configuring CONFIG_HEAP_USE_HOOKS.

How to set the session expiry interval for MQTT?

This feature is only available for MQTT5 protocol, which can be set through `session_expiry_interval` in `esp_mqtt5_disconnect_property_config_t`.

Is the reconnection operation effective after MQTT disconnects? Does it need optimization?

MQTT will automatically reconnect by default, so the application layer does not need to manually disconnect and then reconnect.

What is the cause of the "mqtt_client: No PING_RESP, disconnected" error?

This error usually occurs because the MQTT heartbeat response from the peer was not received within the specified time. It is recommended to use packet capture and log comparison to analyze whether the issue is related to the server or WiFi packet transmission.

Why does the TCP receive window decrease by 2 bytes each time when MQTT connection only has heartbeat packets?

This may be because the data is stuck in the underlying receive buffer, causing the receive window to decrease gradually. To avoid this situation, ensure that the application calls the `read` function correctly to handle the MQTT ping response.

ESP32 acting as an AP connects to two ESP Stations, both of which have established MQTT connections. Can the AP monitor the MQTT messages?

No, ESP32 acting as an AP is only responsible for routing and forwarding. Since the destination address of the packets sent by the Stations is not the AP itself, the AP only forwards the packet upon receipt and does not listen to or process the messages.

Why does the MQTT connection fail with error `MQTT_CONNECTION_REFUSE_NOT_AUTHORIZED`?

This error typically indicates that the username or password is incorrect. Please check and ensure that you provide the correct `username` and `password`.

How to ensure that MQTT devices can still receive messages published during offline periods after going offline?

Configure the MQTT connection parameters as follows:

- Set `session.disable_clean_session` to `true` in `esp_mqtt_client_config_t` to ensure that the device can still receive QoS 1 or QoS 2 messages after reconnecting.
- When publishing messages using `esp_mqtt_client_publish`, set `retain` to `true` so the server retains the message. This allows the device to retrieve the message again after waking up.

Will the MQTT broker send a Last Will message if the client disconnects unexpectedly?

If the client disconnects without sending an MQTT DISCONNECT message and the keepalive interval is short, the broker may assume the client is offline and trigger the Last Will message. It is recommended to explicitly call the disconnect API and set an appropriate keepalive interval.

During MQTT5 operation, frequent Wi-Fi network switching causes repeated re-subscriptions. As a result, `MQTT5 publish check fail` messages flood the log and the connection cannot recover. What is the reason, and how can this be resolved?

MQTT5 provides a flow control mechanism that allows the server to limit the client's message publishing rate. The number of messages that a client can send simultaneously is determined by the server. When the PUBLISH messages sent by the client have not yet received PUBACK confirmation from the server, and the total number reaches the limit set by the server, the client must wait for the server to acknowledge one packet before it can send another. If the device is restricted by flow control, it must wait for the PUBACK acknowledgment before continuing to publish; otherwise, message sending will fail.

In ESP-IDF, is it possible to have MQTT use a specified netif (such as Ethernet), while other services use different netifs?

Yes. This can be achieved by configuring the `if_name` field in `esp_mqtt_client_config_t` to bind to a specific netif. The netif name can be obtained through `esp_netif_get_netif_impl_name(netif, ifr.ifr_name)`. If you need to iterate through all netifs, you can use `netif = esp_netif_next(netif)`.

4.5.6 Other Protocols

How to optimize communication latency for ESP32?

- It is recommended to turn off the sleep function for Wi-Fi by calling the API `esp_wifi_set_ps(WIFI_PS_NONE)`.
- You can also disable the `AMPDU` function in menuconfig.

Does ESP8285 support CCS (Cisco Compatible eXtensions)?

No, it doesn't.

Does ESP32 support LoRa (Long Range Radio) communication?

No, the ESP32 itself does not have the LoRa protocol stack and the corresponding RF parts. However, to realize communication between Wi-Fi and LoRa devices, you can connect an external chip integrated with LoRa protocol to ESP32. In this way, ESP32 can be used as the master control MCU to connect the LoRa chip.

After calling `esp_netif_t* wifiAP = esp_netif_create_default_wifi_ap()` for ESP32-S2 chips, a following call of `esp_netif_destroy(wifiAP)` to deinit caused a 12-byte of memory leakage. What is the reason?

- It is necessary to call `esp_wifi_clear_default_wifi_driver_and_handlers(wifiAP)` before `esp_netif_destroy(wifiAP)`. This is the correct deinit process. Following this process, there will be no memory leakage.
- Alternatively, call `esp_netif_destroy_default_wifi(wifiAP)`, which is supported by ESP-IDF v4.4 and later versions.

How to implement the certificate auto-download function?

CHIP: ESP32

Please refer to aws certificate automatic download function .

How to get more debug information based on errno in ESP-IDF?

- The errno list in ESP-IDF v3.x exists directly in the IDF. Click errno.h to check it.
- The `errno.h` for ESP-IDF v4.x is located under the compiler toolchain. For example, for esp-2020r3, the path of `errno.h` is `/root/.espressif/tools/xtensa-esp32-elf/esp-2020r3-8.4.0/xtensa-esp32-elf/xtensa-esp32-elf/include/sys/errno.h`.

Does the ESP8266_RTOS_SDK support the TR-069 protocol?

No.The ESP8266_RTOS_SDK itself does not provide native support for the TR-069 protocol, but you can implement TR-069 protocol stack according to your requirements and integrate it into the ESP8266_RTOS_SDK. You can also use third-party TR-069 protocol stacks for integration. In summary, ESP8266_RTOS_SDK can support the TR-069 protocol, but requires yourself to integrate and implement.

Does the ESP32 support SAVI?

No, SAVI (Source Address Validation Improvements) is to establish a binding relationship based on IPv6 source address, source MAC address and access device port on the access device (AP or switch) by listening to control packets (such as ND, DHCPv6), i.e. CPS (Control Packet Snooping), and then perform source address validation on IP packets passing through the specified port. Only when the source address of the message matches with the binding table entry can it be forwarded to ensure the authenticity of the source address of data messages on the network. This is generally a policy protocol for switches or enterprise-class AP routers. Currently ESP32 supports IPv6 link-local address and global address for communication.

When Ethernet and Wi-Fi coexist, does Ethernet take precedence over Wi-Fi in data transfer?

:CHIP: ESP32 :

- Call esp_netif_get_route_prio first to check the priority of Ethernet and Wi-Fi. If Wi-Fi takes priority over Ethernet, you can prioritize them by modifying `route_prio` in the structure `esp_netif_t`.

When using SNTP, I encountered the error "assert failed: sntp setoperatingmode IDF/components/lwip/lwip/src/apps/sntp/sntp.c:724 (Operating mode must not be set while SNTP client is running) ?" . How can I solve this issue?

- You can solve it by adding the following code snippet to the sntp_init() interface:

```
if(sntp_enabled()){
sntp_stop();
}
```

Have the ESP modules been IPv6 certified?

- Currently not.

Does the ESP module support the IPP protocol for printers?

Currently not supported.

When ESP32 connects to an open hotspot that requires login authentication (such as CMCC), how to handle this situation?

This type of hotspot typically uses the `Captive Portal` mechanism. When ESP32 connects to such a hotspot, the device will be redirected to a login page where the user needs to enter the mobile number and verification code for authentication. Currently, there is no ready-made solution to automate this authentication process. An alternative approach is to add the device to the whitelist on the router' s management page, thus avoiding the need for authentication each time it connects.

What is the time interval for switching NTP servers when using the NTP function? Is there an interface to change the time of NTP switching?

After the first NTP server fails, the interval time starts to increase from 15 seconds and eventually stabilizes at about two and a half minutes. Currently, there is no interface to modify the switch time of the NTP server.

Why is the sign of the timezone in SNTP opposite to the actual situation? For example, the timezone in India is GMT+5:30, but in the code, it has to be written as GMT-5:30?

This is the design of the standard time API. The plus and minus signs indicate the direction of the time zone offset relative to GMT, using the GMT offset to represent the local time zone.

What should be noted about the handling logic for send timeouts?

Regarding the handling logic for send timeouts, if error code 11 (indicating too fast sending) is treated as an error, but it does not actually constitute a real exception, then it should not directly trigger an exception handling process that terminates the transmission. The correct approach is to allow quick retries within a certain range, rather than immediately terminating the connection.

Under normal network connection, why does repeated send timeout occur with error code 113?

Error code 113 usually indicates that the underlying link has been disconnected or that TCP retransmissions have reached their maximum limit. Packet capture is needed to verify whether the device successfully sent the packets and whether the peer returned a TCP ACK.

Does ESP-IDF support the Undefined Behavior Sanitizer (UBSAN) tool?

ESP-IDF supports UBSAN, and you can refer to the official documentation: Undefined Behavior Sanitizer Checks.

How to prevent ICMP packet attacks?

Currently, there is no specific protection mechanism in IDF to prevent ICMP message attacks. It is recommended to either disable the ICMP function, or limit the processing frequency of ICMP messages, to reduce the impact of attacks on the system.

What could be the reason for an assertion failure during SNTP initialization (*sntp_init*)?

Repeated initialization causes assertion failure, ensure that SNTP is only initialized once. Refer to the fix: GitHub Issue #12854.

What are the encryption suite requirements for European certification?

European certifications (such as EN18031) require the use of encryption suites with strong security. Weak encryption suites (such as SHA1, MD5, DES, and RC4) are prohibited. It is recommended to prioritize R-class encryption suites and use ciphersuite.info to evaluate the security of the encryption suites.

4.6 Provisioning

4.6.1 Can I add any broadcast data I want to Android ESP-Touch (e.g., add a device ID so that ESP32 can receive this ID)?

- ESP-Touch is a communication protocol used to establish a Wi-Fi connection between a mobile phone and ESP8266/ESP32. It uses a special broadcast method to transmit Wi-Fi SSID and password. Typically, the data broadcasted by ESP-Touch should be in a fixed format and cannot include customized data.
- If you want ESP32 to receive customized data such as device ID, you can consider using other communication protocols such as MQTT or HTTP. With these protocols, you can define data formats as you wish and communicate between an Android App and ESP32.
- If you still want to broadcast customized data, you can use BluFi, which is the networking protocol based on Bluetooth LE. Please refer to the following references for BluFi:
  - Android APP: https://github.com/EspressifApp/EspBlufiForAndroid.
  - iOS APP: https://github.com/EspressifApp/EspBlufiForiOS.

4.7 Security

4.7.1 Is the firmware in ESP8266 readable?

Yes, because the firmware in ESP8266 is located in the external flash, thus can be read externally. In addition, ESP8266 does not support flash encryption and all the data is written in plaintext.

4.7.2 Is it possible to encrypt firmware for ESP8285?

- No, the ESP8285 chip does not support firmware encryption function.
- Both ESP32 and ESP32-S2 support firmware encryption, thus can be your substitution.
- If you insist on using ESP8285, you can achieve data encryption by adding an encrypted chip externally.

4.7.3 What is the difference between secure boot v1 and v2?

Compared with secure boot v1, secure boot v2 has the following improvements: - The bootloader and app use the same signature format. - The bootloader and app use the same signing key.

Currently, secure boot v1 is only reommended for earlier versions than ESP32 v3.0. For ESP32 v3.0 and later versions, ESP32-C3, ESP32-S2, and ESP32-S3, it is recommended to use secure boot v2.

4.7.4 After enabling secure boot, there is a build error indicating missing files. What could be the reasons ?

Error log: /Makefile.projbuild:7/f/ESP32Root/secure_boot_signing_key.pem

Reason: security boot is a function for firmware signature verification, which requires generating key pairs. - For the method of generating key pairs when enabling secure boot v1, please refer to secure boot v1 generate key. - For the method of generating a key pair when secure boot v2 is enabled, please refer to secure boot v2 key generation.

4.7.5 After enabling secure boot, is it possible for modules to be flashed again?

- If the secure boot v1 is configured as one-time, then it can only be flashed once and the bootloader firmware cannot be reflashed.
- If the secure boot v1 is configured as reflashable, then the bootloader firmware can be flashed again.
- The secure boot v2 allows reflashing the bootloader and app firmware.

4.7.6 With flash encryption enabled, a module reports an error as `flash read error` after reflashed. How to resolve such issue?

With flash encryption enabled, the module will not support plaintext firmware flash. For common failures, please refer to Possible Failures. You can use the espefuse script to disable the encryption and then reflash the plaintext firmware, or directly flash the encrypted firmware to devices referring to the flash encryption example.

Note: Please note there is a time limit for the flash encrypted function.

4.7.7 After enabling flash encryption and secure boot for ESP32, how to disable them?

- If you are using the one-time flash (Release) mode, both flash encryption and secure boot cannot be disabled.
- If you are using the reflashable (Development (NOT SECURE)) mode, the flash encryption can be disabled, please refer to Disabling Flash Encryption; while the secure boot cannot be disabled.

4.7.8 Is there any security strategy for ESP32 to protect its firmware?

- ESP32 supports flash encryption and secure boot.
- For flash encryption, please refer to flash encryption.
- For secure boot, please refer to secure boot.
- For secure boot V2, please refer to secure boot V2 for chip revision v3.0.

4.7.9 When ESP32 debugging GDB after enabling flash encryption, why does it continuously reset and restart?

- After ESP32 enabling flash encryption or secure boot, it will restrict JTAG debugging by default, please refer to Tips and Quirks.
- You can read the current JTAG status of your chip using the `espefuse.py summary` command from esptool.

4.7.10 How to enable flash encryption for ESP32?

- It can be enabled via menuconfig or idf.py menuconfig by configuring `Security features` -> `Enable flash encryption on boot (READ DOCS FIRST)`.
- Please refer to Flash encryption instructions.

4.7.11 After GPIO0 is pulled down, the ESP32 cannot enter download mode and prints "download mode is disable" . What could be the reason?

- The log means the chip' s UART Download mode has been disabled. You can check this via the `UART_DOWNLOAD_DIS` bit in eFuse.
- Please note that after the Production mode of flash encryption is enabled, the UART Download mode will be disabled by default. For more information, please refer to UART ROM download mode.

4.7.12 Can the secure boot function be enabled for ESP32 in Arduino development environment?

- No. If you want to use Arduino for development, the only way to enable the secure boot function is to use Arduino as an IDF component.

4.7.13 What are the use scenarios for secure boot and flash encryption?

- When secure boot is enabled, the device will only load and run firmware that is signed by the specified key. Therefore, it can prevent the device from loading illegal firmware and prevent unauthorized firmware from being flashed to the device.
- When flash encryption is enabled, the partitions on the flash where firmware is stored and the data in the partitions marked as "encrypeted" will be encrypted. Therefore, it can prevent the data from being illegally viewed, and firmware data copied from flash cannot be applied to other devices.

4.7.14 What are the data stored in eFuse involved in secure boot and flash encryption?

- For the storage used in secure boot v1, please refer to secure boot v1 efuses.
- For the data stored in eFuse used in secure boot v2, please refer to secure boot v2 efuses。
- For the data stored in eFuse used in flash encryption, please refer to flash encryption efuses。

4.7.15 Enabling secure boot failed with the log "Checksum failure". How to fix it?

- After enabling secure boot, the size of bootloader.bin will increase, please check whether the size of the bootloader partition is enough to store the compiled bootloader.bin. For more information, please refer to Bootloader Size。

4.7.16 NVS encryption failed to start and an error occurred as `nvs: Failed to read NVS security cfg: [0x1117] (ESP_ERR_NVS_CORRUPT_KEY_PART)`. How can I solve this issue?

- Please erase flash once using the flash tool before starting NVS encryption, and then flash the firmware which can enable the NVS encryption to the SoC.

4.7.17 After flash encryption was enabled, a warning occurred as `esp_image: image at 0x520000 has invalid magic byte (nothing flashed here)`. How can I solve this issue?

- After SoC starts flash encryption, it will try to encrypt the data of all the partitions of the app type. If there is no corresponding app firmware stored in one app partition, the above log will appear. To avoid this warning, you can flash pre-compiled app firmware to the partitions of the app type when starting flash encryption.

4.7.18 Why is reltead data not encrypted after I enable `CONFIG_EFUSE_VIRTUAL` and flash encryption?

- Currently, Virtual eFuses is only used to test the update of eFuse data. Thus, flash encryption is not enabled completely even this function is enabled.

4.7.19 Can I update an app firmware which enables flash encryption in a device which does not enable fash encryption through OTA?

- Yes, please deselect `Check Flash Encryption enabled on app startup` when compiling.

4.7.20 How can I delete keys of secure boot?

- Keys of secure boot should be deleted in the firmware `new_app.bin`. First, please assure that `new_app.bin` is employed with two signatures. Then, flash `new_app.bin` to the device. At last, when the original signatures are verified, you can delete the original keys through `esp_ota_revoke_secure_boot_public_key()` in `new_app.bin`. Please note that if you use the OTA rollback scheme, please call `esp_ota_revoke_secure_boot_public_key()` after `esp_ota_mark_app_valid_cancel_rollback()` returns `ESP_OK`. For more details, please refer to Key Revocation.

4.7.21 After I enabled secure boot or flash encryption (development mode), I cannot flash the new firmware, and an error occurred as `Failed to enter Flash download mode`. How can I solve this issue?

- Generally, the above log indicates that your flash command is incorrect. Please use script `idf.py` to execute `idf.py bootloader` and `idf.py app` to compile `bootloader.bin` and `app.bin`. Then execute the flash command through `idf.py` according to the tips after compiling. If you still cannot flash your firmware, please use `espefuse.py -p PORT summary` to check the eFuse of the current device and check whether the flash download mode is enabled or not.

4.7.22 After I input the command `espefuse.py read_protect_efuse BLOCK3` command in the terminal configured with ESP-IDF to enable the read-protection for Efuse BLOCK3, why is the data of the Efuse BLOCK3 all 0x00 when I input `esp_efuse_read_block()` to read the Efuse BLOCK3?

- After the Efuse BLOCK3 is read protected, it cannot be read anymore.

4.7.23 How can I enable secure boot or flash encryption by pre-burning eFuse?

By default, you can enable secure boot or flash encryption by burning firmware with secure boot or flash encryption enabled. In addition, you can also enable secure boot or flash encryption by pre-burning eFuse in the following two methods: - With flash_download_tool, eFuse will be pre-burned automatically if secure boot or flash encryption is enabled. - You can generate the key and burn corresponding eFuse blocks with espsecure.py and espefuse.py.

4.7.24 After enabling Secure Boot, why can't the new bootloader.bin be flashed using the `idf.py flash` command?

- After enabling Secure Boot, compile the new bootloader.bin using the `idf.py bootloader` command. Then, flash the new bootloader.bin using the `idf.py -p (PORT) bootloader-flash` command.
- In ESP-IDF v5.2 and later versions, you can also solve this problem by enabling the `CONFIG_SECURE_BOOT_FLASH_BOOTLOADER_DEFAULT` option. For details, please refer to CONFIG_SECURE_BOOT_FLASH_BOOTLOADER_DEFAULT.

4.7.25 After enabling Secure Boot or flash encryption, how can I view the security-related information in the device?

Please use the command *esptool.py –no-stub get_security_info* to view the security information of the device.

4.7.26 After enabling Secure Boot or flash encryption, what should I pay attention to during OTA (Over-The-Air) updates?

- After enabling Secure Boot, you must sign the new firmware to be used for OTA updates. Otherwise, the new firmware cannot be applied to the device.
- After enabling flash encryption, when generating a new firmware, please ensure that the flash encryption option is enabled.

4.7.27 Which USB functions will be disabled after the ESP32-S3 enables flash encryption or Secure Boot?

- After enabling flash encryption or secure boot on ESP32-S3, the USB-JTAG debugging function will be disabled, and it does not support burning firmware with the idf.py dfu-flash command via the USB interface.
- After enabling flash encryption or secure boot on ESP32-S3, it supports USB Host and USB Device features; it also supports downloading firmware through the USB interface using the `idf.py flash` command.

4.7.28 After enabling flash encryption, if there are multiple flash encryption keys for `XTS_AES_128_KEY` in the device's eFuse, how will the device select the key?

- The device will always choose the key with the smallest `Key ID`.

4.7.29 When Secure Boot V2 is enabled, how can I store the public key used for signature verification on the device?

- The information of public key is stored in the device's signature block. When Secure Boot V2 is initially enabled, the device will automatically read the public key information from the signature block and write it into the device.

4.7.30 After enabling the Secure Boot V2 feature on ESP series products, is it still possible to reflash the firmware?

- After enabling the Secure Boot V2 feature on ESP series products, if the download mode is not disabled, re-flashing of firmware is supported.
- Note: For the ESP series chips, when the Secure Boot V2 function is enabled, the default configuration of the Flash Download Tool does not support firmware reflashing. You need to modify the default configuration in the tool to support firmware reflashing. Take ESP32-C3 as an example:
  - Modify the default configuration in the `esp32c3 > security.conf` file: Change `flash_force_write_enable = False` to `flash_force_write_enable = True`.
  - Modify the default configuration in the `esp32c3 > spi_download.conf` file: Change `no_stub = False` to `no_stub = True`.
  - If using esptool, run the following command to reflash the firmware:

```
esptool.py --chip esp32c3 -p COM68 -b 460800 --before=default_reset --
↪after=no_reset --no-stub write_flash --force --flash_mode dio --flash_
↪freq 80m --flash_size keep 0x0 bootloader.bin 0xF000 partition-table.bin␣
↪0x20000 blink.bin
```

4.7.31 Does the flash encryption scheme support encryption for the file system?

It supports encryption on the fatfs file system, but does not support encryption on the spiffs file system.

4.7.32 How to create an encrypted `nvs_data.bin` when using NVS encryption?

- When enabling the NVS encryption scheme, the device will not encrypt the NVS data during flashing, so it is necessary to use the script tool on the PC side to encrypt `nvs_data.bin`.
- After enabling the NVS encryption scheme, the device will automatically encrypt NVS data when executing APIs of the `nvs_set_*` type; it will automatically decrypt NVS data when executing APIs of the `nvs_get_*` type.

4.7.33 Does the SPIFFS file system based on ESP32 support the flash encryption scheme?

No. The internal structure of SPIFFS does not support integration with flash encryption. If you need a file system that supports

---

4.7.34 Using the ESP32-C3 SDK based on ESP-IDF v5.0.6, NVS encryption based on Flash Encryption is enabled in the software configuration. After the device completes Flash Encryption and restarts the firmware, the firmware runs with the following error. What is the reason?

```
I (438) main_task: Calling app_main()
E (438) nvs: Failed to read NVS security cfg: [0x1117] (ESP_ERR_NVS_
↪CORRUPT_KEY_PART)
ESP_ERROR_CHECK failed: esp_err_t 0x1117 (ESP_ERR_NVS_CORRUPT_KEY_PART)␣
↪at 0x42007e96
0x42007e96: app_main at /home/caiguanhong/esp/esp-idf-5.0.6/esp-idf/
↪examples/wifi/getting_started/softAP/build/../main/softap_example_main.
↪c:95 (discriminator 1)

file: "../main/softap_example_main.c" line 95
func: app_main
expression: ret

abort() was called at PC 0x40386249 on core 0
0x40386249: _esp_error_check_failed at /home/caiguanhong/esp/esp-idf-5.0.
↪6/esp-idf/components/esp_system/esp_err.c:47

Stack dump detected
Core  0 register dump:
MEPC    : 0x40380938  RA      : 0x40386254  SP      : 0x3fc9a260  GP      ␣
↪: 0x3fc91400
0x40380938: panic_abort at /home/caiguanhong/esp/esp-idf-5.0.6/esp-idf/
↪components/esp_system/panic.c:425

0x40386254: __ubsan_include at /home/caiguanhong/esp/esp-idf-5.0.6/esp-
↪idf/components/esp_system/ubsan.c:313
```

- When using NVS Encryption: Flash Encryption-Based Scheme, it is necessary to thoroughly erase the nvs_keys partition before starting the application. Otherwise, the application may generate an *ESP_ERR_NVS_CORRUPT_KEY_PART* error code.
- Before downloading the firmware, please use the *idf.py erase-flash* command to erase the flash.

4.7.35 How to implement firmware signature verification (ECDSA) at the application level?

For application-layer verification, refer to the Bootloader implementation in secure_boot_signatures_app.c#L56.

4.7.36 How to perform encryption on custom data?

You can use the APIs provided by mbedtls to perform encryption, checksum calculation, signature generation, and other operations on custom data. For an example of AES-based data encryption and decryption, refer to components/mbedtls/test_apps/main/test_aes.c.

4.7.37 What is the difference between the NVS encryption scheme based on the Flash encryption framework and the NVS encryption scheme based on HMAC?

- The NVS encryption scheme based on the Flash encryption framework requires the Flash encryption framework to be enabled first. The NVS encryption key is stored in the partition table. When it is necessary to protect the contents of the app partition, this scheme is recommended.
- The HMAC-based NVS encryption scheme does not require the activation of the Flash encryption framework. The NVS encryption key is stored in the eFuse special memory. This scheme is recommended when only the contents of the NVS partition need to be protected.

4.8 Storage

4.8.1 FAT Filesystem

How to improve the damage to FatFs file system caused by accidental power loss?

Since FatFs is designed to not support write transactions, the accidental power loss may cause error to partitions, which cannot be restored by simply modifying FatFs. For now, it is recommended to resolve this problem in application level by creating two identical FatFs partitions to do backups, or you can also choose a more secure file system instead, such as LittleFS and SafeFAT (charged).

How to make and flash the image of a FatFs file system?

Here we will use a third-party tool, since there is no such tool provided in ESP-IDF now. The entire process shows as below:

- Step 1: Use the mkfatfs tool to create an image from a specified folder, for example, create an image named fat_img.bin with a size of 1048576 Bytes from the file_image folder:

```
./mkfatfs -c file_image -s 1048576 ./fat_img.bin
```

- Step 2: Flash the image, for example, flash it to 0x110000:

```
esptool.py -p /dev/ttyUSB1 -b 460800 --before default_reset --after hard_
↪reset write_flash --flash_mode dio --flash_size detect --flash_freq 80m␣
↪0x110000 ~/Desktop/fat_img.bin;
```

- Step 3: mount the image in program:

```
static void initialize_filesystem() {
  static wl_handle_t
  wl_handle = WL_INVALID_HANDLE;
  const esp_vfs_fat_mount_config_t
  mount_config = { .max_files = 10, };
  ESP_LOGI(TAG, "Mounting FATfilesystem");
  esp_err_t err = esp_vfs_fat_spiflash_mount("/spiflash", "storage", &
↪mount_config, &wl_handle);
  if (err != ESP_OK) {
      ESP_LOGE(TAG, "Failed to mount FATFS (%s)", esp_err_to_name(err));
      return;
  }
}
```

Note: The address to be flashed in step 2 must be the corresponding partition address in the partition table, and the created image must be the same size as the one set in the partition table. Please remember to go to menuconfig and set `Component config` > `Wear Levelling` > `Wear Levelling library sector size` to 512, otherwise it will cause a mounting failure.

What are the differences between the two file systems, FatFs and SPIFFS, and how do we choose?

Please refer to *File System <https://github.com/espressif/esp-iot-solution/blob/master/docs/en/storage/file_system.rst>*.

What is the maximum size supported by FatFs?

The FAT32 file system supports a maximum capacity of 2 TB. However, due to the limitations of some operating systems (such as Windows), FatFs is generally only used on storage devices up to 32 GB. For storage devices larger than 32 GB, file systems like exFAT are typically used by default.

I cannot open the files with long names when I use the FAT file system. How can I fix this issue?

- You can modify this in `menuconfig` > `Component config` > `FAT Filesystem support` > `Long filename support`, and choose either the `Long filename buffer in heap` or `Long filename buffer on stack`. Then, you can change the maximum filename length in `Component config` > `FAT Filesystem support` > `Max long filename length`.

When I used the ext_flash_fatfs example to test, I encountered an error `vfs_fat_spiflash :f_mks failed(14),config:Failed to mount FATFS(ESP_FAIL)` if I set the fatffs partition to less than 512 KB. How can I solve it?

A FAT partition has 128 sectors at the minimum, so the minimum size of the file system should be 128*4+4*4=528 KB. The extra four sectors are used for wear leveling information. As a result, the size of the fatffs partition must not be less than 528 KB.

How to support CJK (Chinese, Japanese, Korean) or other language filenames in FATFS?

FatFs supports Unicode filenames. To manually adjust the multilingual support for filenames, please modify the following options in *menuconfig*:

- Navigate to *(Top)* > *Component config* > *FAT Filesystem support* > *OEM Code Page*, and set it to *Dynamic (all code pages supported) (FATFS_CODEPAGE_DYNAMIC)*. With this setting, FATFS will support all code pages, but it will increase the size of the compiled output by about 500 kB. If you choose another code page, make sure that the selected code page matches the character set used in the filenames; otherwise it may cause garbled text, file not found errors, or other latent issues.
- Navigate to *(Top)* > *Component config* > *FAT Filesystem support* > *API character encoding-UTF-8 encoding (FATFS_API_ENCODING_UTF_8)*. This option controls the encoding method used by the filenames read by the FATFS API. For more details, please refer to the configuration help.

4.8.2 Non-Volatile Storage (NVS)

If data needs to be stored or updated to flash every minute, can ESP32 NVS meet this requirement?

According to NVS Specifications, NVS uses two main entities in its operation: pages and entries. Logical page corresponds to one physical sector of flash memory. For now, we assume that flash sector is 4096 bytes and each page can contain 126 entries (32 bytes for one entry), with the left spaces for page header (32 bytes) and entry state bitmap (32 bytes). Typical flash lifetime is 100 k erase cycles. Assuming that the device is expected to run for 10 years, and the data size written to flash is 4 bytes per minute with flash encryption disabled, then the number of flash write operation can be calculated as: 60×24×365×10=5256000. In this way, no more than 42 k of erase cycles (5256000/126) will be caused in NVS, which does not exceed 100 k. Therefore, such operation is supported even without the effects of multiple sectors. In actual use, usually there will be multiple sectors given to NVS, and the NVS can distribute erase cycles to different sectors, making the number of erase cycles in each sector necessarily less than 42 k.

Therefore, theoretically, NVS can meet the erasure requirements. Please note that the lifespan of different flash models may vary, which should be evaluated based on the actual situation.

Is it feasible to perform 10,000,000 erase/write operations each time when writing data of 1 KB or less?

If each write operation involves 1 KB of data, and the smallest erasable unit of flash is a 4 KB page , then a single erase block can be considered as containing four logical addresses of 1 KB each. Therefore, 10,000,000 write operations are equivalent to 2,500,000 erase/write cycles for a single page (10,000,000 * 1 KB / 4 KB). To avoid exceeding the erase/write cycle limit for each physical block (typically 100,000 cycles), at least 25 physical 4 KB blocks are required to distribute these operations evenly (2,500,000 cycles / 100,000 cycles per block = 25 blocks).

Considering that the actual available space in each sector may be reduced due to system overhead, and in order to improve the reliability and stability of the system, it is recommended to reserve more physical space. For example, you can allocate about 150 KB of space to ensure sufficient margin to accommodate extra system requirements and future scalability.

Does NVS have wear levelling function?

Yes, NVS (Non-Volatile Storage) has wear leveling functionality. When storing data in flash memory, the limited number of writing and erasing operations to the flash will cause some storage blocks to have shorter service life than others, affecting the overall working life of the memory. To address this issue, NVS uses an erase-write balancing mechanism implemented internally, rather than the wear_levelling component in ESP-IDF, to evenly distribute data across the flash memory blocks, ensuring that each block is used as much as possible to extend the overall working life of the flash memory.

See more information in Wear Levelling API.

Can NVS sectors be corrupted by accidental power loss during writing?

No, NVS is designed to resist accidental power loss, so it will not be damaged.

Will the configured Wi-Fi SSID and PASSWORD disappear after the ESP series development board is powered on again and need to be reconfigured?

- It will be stored in NVS by default and will not disappear due to power failure. You can also set it through `esp_wifi_set_storage()`, which can be divided into two situations:
  - If you want to save the Wi-Fi SSID and PSAAWORD when powered off, you can store the Wi-Fi information in flash by calling `esp_wifi_set_storage(WIFI_STORAGE_FLASH)`.
  - If you want to achieve the operation of not saving the Wi-Fi SSID and PASSWORD when powered off, you can call `esp_wifi_set_storage(WIFI_STORAGE_RAM)` to store the Wi-Fi information in RAM.

How to realize that stored user data can be saved after power off, not erased by OTA, and can be re-written or modified?

- Non-volatile storage can meet the above requirements, and only eFuse and flash can be used to store the data. Since the data should be modifiable, only flash is suitable. It is recommended to use NVS or MFG mechanism. For details, please refer to
  - Manufacturing Utility
  - NVS Partition Generator Utility

Is the API provided by the NVS library thread-safe?

- The NVS initialization and de-initialization functions are not thread-safe, but other NVS functions (open/get/set/commit/close) are thread-safe.

What are the meanings of NVS error codes in the ESP-IDF SDK?

- For the meanings of NVS error codes in the ESP-IDF SDK, refer to esp-idf/components/nvs_flash/include/nvs.h.

Is the cache disabled when writing data to NVS?

- When writing data to NVS, the cache will be disabled, and no functions that operate flash can be called within the interrupt function.

4.8.3 PSRAM

When using ESP32 modules, how to check the size of their PSRAM?

For ESP32 modules, the esp_spiram_get_size() function in ESP-IDF can be used to obtain the size of a module's PSRAM. This function returns the total size of the PSRAM in bytes and can be used for memory allocation and management.

The following is an example for obtaining the size of PSRAM:

```
#include "esp_psram.h"
...
size_t psram_size = esp_psram_get_size();
printf("PSRAM size: %d bytes\n", psram_size);
```

Note that the esp_spiram_get_size() function should be called before using the PSRAM to ensure the correct PSRAM size can be obtained. Additionally, PSRAM functionality should be enabled in `make menuconfig`, so that PSRAM can be used and configured. Furthermore, the PSRAM size can also be obtained in the bootloader log.

When ESP32 connected to a PSRAM externally, how to change its clock source?

Modify in menuconfig:

- ESP-IDF v4.x: `menuconfig` > `Component config` > `ESP32-specific` > `SPI RAM config`。
- ESP-IDF v5.x: `menuconfig` > `Component config` > `ESP PSRAM` > `SPI RAM config`。

When a 8 MB PSRAM mounted on ESP32, why only 4 MB of it is actually mapped?

- It is recommended to use the official ESP-PSRAM chip in conjunction with the ESP32 chip.
- The maximum capacity of PSRAM supported by different chip series varies. For the ESP32 series, up to 4 MB (0x3F80_0000 to 0x3FBF_FFFF) of the external RAM can map into the data address space. For details, refer to ESP32 Datasheet > section 4.1.4 `Address Mapping Structure`.
- For a 8 MB PSRAM, you can access the other 4 MB following example himem.

I' m using an ESP32 development board with the official PSRAM chip PSRAM64H embedded. But after replacing another type of PSRAM chip to PSRAM64H, it failed to recognize when I ran an ESP-IDF example and enabled the PSRAM configuration. What is the reason?

- To change the model of the PSRAM chip, you need to modify the corresponding configuration options in menuconfig:
  - ESP-IDF v4.x: `menuconfig > Component config > ESP32-specific > Support for external, SPI-connected RAM > SPI RAM config > Type of SPI RAM chip in use`
  - ESP-IDF v5.x: `menuconfig > Component config > ESP PSRAM > Support for external, SPI-connected RAM > SPI RAM config > Type of SPI RAM chip in use`
- If you cannot find the corresponding type options of the new PSRAM chip you are about to use, please add the chip driver manually.

Why is the following error printed when I download the hello-world example into the ESP32-WROOM-32E module?

```
E (225) psram: PSRAM ID read error: 0xffffffff
E (225) spiram: SPI RAM enabled but initialization failed. Bailing out.
```

The error is due to: The PSRAM setting is enabled in the software (ESP-IDF v4.x path: `Component config > ESP32-specific > Support for external, SPI-connected RAM`; ESP-IDF v5.x path: `Component config > ESP PSRAM > Support for external, SPI-connected RAM`), but no available PSRAM was detected on the hardware.

Does ESP32 support coexistence between 16 MB External Flash and 8 MB External PSRAM?

Yes, ESP32 supports coexistence between 16 MB External Flash and 8 MB External PSRAM.

After enabling the `BT/BLE will first malloc the memory form the PARAM` configuration option on ESP32-S3-WROOM-N4R2, the software prints the following error log. However, it runs normally after disabling the Bluetooth LE 5.0 configuration. Why?

```
E(36997)BLE_INIT:Mallocfailed
E(37307)BLE_INIT:Mallocfailed
E(38307)BLE_INIT:Mallocfailed
E(39307)BLE_INIT:Mallocfailed
E(40307)BLE_INIT:Mallocfailed
```

- The current error is due to insufficient Malloc memory. When the application memory is less than the `Maximum malloc() size, in bytes, to always put in internal memory` configuration, the internal memory of the chip will be used by default. The path to this configuration is:
  - ESP-IDF v4.x: `idf.py menuconfig > Component config > ESP32-specific > Support for external, SPI-connected RAM > SPI RAM config`
  - ESP-IDF v5.x: `idf.py menuconfig > Component config > ESP PSRAM > Support for external, SPI-connected RAM > SPI RAM config`

  You can reduce this configuration, or change the `SPI RAM access method` configuration to `Make RAM allocatable using heap_caps_malloc(..... MALLOC_CAP_SPIRAM)`.

Does ESP32-C6 support external PSRAM?

- ESP32-C6 does not support external PSRAM, but ESP32-C61 supports.

When developing with the ESP32-PICO-V3-02 chip on ESP-IDF v5.1.2, does the PSRAM speed only support 40 MHz?

- PSRAM speed also supports 80 MHz. To set it to 80 MHz, please update the configuration `idf.py menuconfig > Serial flasher config > Flash SPI speed` to 80 MHz.
- Typically, we recommend the software settings of 80 MHz flash speed + 80 MHz PSRAM speed.

When using the xTaskCreateWithCaps() API to allocate external PSRAM, I encountered the following error. Why?

```
assert failed: xTaskCreateStaticPinnedToCore freertos_tasks_c_additions.
↪h:314 (xPortcheckValidStackMem(puxStackBuffer))
```

When using `xTaskCreateWithCaps()` to allocate PSRAM, you need to enable the `Support for external, SPI-connected RAM` configuration in menuconfig (ESP-IDF v4.x path: `Component config > ESP32-specific`; ESP-IDF v5.x path: `Component config > ESP PSRAM`), then set `SPI RAM config > SPI RAM access method` to `(X) Make RAM allocatable using malloc() as well` mode, and finally, you need to enable the `[*] Allow external memory as an argument to xTaskCreateStatic` configuration option.

4.8.4 SD/SDIO/MMC Driver

Is it possible to use ESP8266 together with TF card?

It is not recommended to use ESP8266 together with TF card.

- Although ESP8266 can be connected to TF card in hardware level (communicate through SPI), the chip may run out of memory in different application scenarios due to its limited resources. Thus, it is not recommended to use ESP8266 with TF card.
- If all you need is a Wi-Fi-only module that can be connected to a TF card, we recommend using chips like ESP32-S2.

What is the maximum capacity of eMMC supported by ESP32-S3?

2 TB is the limit for SD and eMMC protocols. There is no specific maximum capacity limit for the ESP32-S3. But if your application is built upon a filesystem, the maximum capacity may also be restricted by the filesystem.

Does ESP32 support DDR52, HS200, HS400, and SDR52 modes when it is connected to an eMMC card?

- At the frequency of 40 MHz, ESP32 supports the DDR52 mode and does not support other modes. For details, please refer to Supported Speed Modes.

Does ESP32-C5/C6 support Quad SPI mode when accessing an SD card via the SPI interface?

- Not supported. The SDSPI application only supports standard SPI mode, with a maximum clock frequency of 40 MHz.
- You can increase the standard SPI rate as follows:
  - Use SPI IO MUX pins
  - Use the maximum supported clock frequency
  - Shorten the length of SPI pin wiring
  - Increase task buffer size
  - Enable DMA mode
  - Place SPI ISR into IRAM

4.8.5 SPI Flash

What is the requirement for the storage and usage of ESP32 flash?

The external flash can be mapped into CPU instruction space and RO data space simultaneously. ESP32 can support up to 16 MB of external flash.

- When it is mapped into CPU instruction space, up to 11 MB + 248 KB of data can be mapped at a time. If more than 3 MB + 248 KB is mapped at a time, the cache performance may be degraded due to speculative CPU reads.
- When it is mapped into RO data space, up to 4 MB of data can be mapped at a time, and 8-bit, 16-bit and 32-bit reads are supported.
- You should specify flash partition table when programming. Specifically, you should divide flash into different partitions, such as app partition, data partition, and OTA partition, and you should specify the size and offset address of each partition.
- When using flash, you need to pay attention to its service life. As flash can sustain only a limited number of erase operations, you need to plan and manage the usage of flash. For example, you can use wear leveling to extend the service life of flash.
- It should be noted that writing operations to flash occupy CPU resources, which may influence the system response time. As a result, the writing operation to flash should be avoided as much as possible.

What kind of sectors are reserved for customized use in ESP8266 modules?

- For previous versions of SDK rel3.0, besides for bootloader and app bin, the following sectors are reserved at the end of the configured flash: 1 for system information, 1 for OTA information and 1 for RF calibration information.
- For SDK rel3.0 and later versions, we use partition_table to manage flash. Except for partition_table and bootloader, other bin files are all marked in partition_table.

How to read flash data for ESP8266?

- You can use the script tool under ESP8266-RTOS-SDK to read flash data. The whole process is shown as follows:
  - Install python environment and the required packages;
  - Go to ESP8266_RTOS_SDK/components/esptool_py/esptool;
  - Run `python esptool.py --chip esp8266 --port /dev/ttyUSB0 --baud 115200 read_flash 0x0 0x400000 esp8266.bin`. In this command, "esp8266.bin" is a self-defined file, where all flash data read will be stored; "/dev/ttyUSB0"is the serial port number in linux environment, which can be different in other environments and systems.

Why do different ESP32 modules have inconsistent flash erase times?

- Different ESP32 modules may have different flash chips or flash controllers, which may affect the erase time. Some models of flash do not have a mechanism to skip empty blocks during erasing, so it takes longer time. Specifically, different flash chips may have different erase time, for example, the erase time of SPI flash and QSPI flash is different. Even the same type of flash chip may have different erase time as they are produced and packaged in different batches. In addition, the design and performance of the flash controller may also affect the erase time. Therefore, erase time varies in ESP32 modules with different flash chips and flash controllers.

When the flash SPI mode is set to QIO mode on the ESP32-S3-WROOM-2-32R8V module, the running firmware prints the following error. Why?

```
E (47) qio_mode: Failed to set QIE bit, not enabling QIO mode
```

The ESP32-S3-WROOM-2-32R8V module uses the 32 MB Octal flash and the 8 MB Octal PSRAM. Please enable the settings in the configuration options:

- `(Top) > Serial flasher config > [*] Enable Octal Flash > Flash SPI mode (OPI)`
- `(Top) > Component config > ESP PSRAM > Support for external, SPI-connected RAM > SPI RAM config > Mode (QUAD/OCT) of SPI RAM chip in use`

How can I confirm whether ESP-IDF supports a certain flash?

- You can refer to Optional features for flash for further understanding of the flash information supported by ESP-IDF. Please note that this document only indicates that the ESP-IDF code supports these flash features, and it is not a list of stable flash certified by Espressif.
- For further support on flash selection, please contact Espressif.

For the SPI flash connected to ESP32-S3, what is the maximum amount of data that can be written in a single operation?

- Due to hardware limitations, ESP32-S3 allows a maximum of 64 bytes of data per operation.

The following error occurs when setting up an OTA partition for firmware upgrades on a flash area exceeding 16 MB, using ESP32-S3R8 with an external GD25Q256EYIGR (32 MB) quad-SPI flash. However, the same test succeeds on the ESP32-S3-WROOM-2-N32R8 module. What could be the reason?

```
E (136196) esp_image: Checksum failed. Calculated 0xe0 read 0x2
E (136196) esp_ota_ops: New image failed verification
E (136206) simple_ota_example: Firmware upgrade failed
```

Additionally, if a factory app partition is directly placed beyond the 16 MB flash boundary to store firmware, the following error occurs:

```
I (52) boot: Partition Table:
I (55) boot: ## Label            Usage          Type ST Offset  ␣
↪Length
I (61) boot:  0 nvs              WiFi data        01 02 00009000␣
↪00006000
I (68) boot:  1 phy_init         RF data          01 01 0000f000␣
↪00001000
I (74) boot:  2 factory          factory app      00 00 01000000␣
↪00100000
I (81) boot: End of partition table
I (84) esp_image: segment 0: paddr=01000020 vaddr=3fce2820␣
↪size=014f0h (  5360) load
E (91) esp_image: Segment 0 0x3fce2820-0x3fce3d10 invalid:␣
↪overlaps bootloader stack
E (99) boot: Factory app partition is not bootable
E (103) boot: No bootable app partitions in the partition table
```

- To achieve full support (including code execution and data access) in a quad flash area exceeding 16 MB, please enable the following experimental configuration options:
  - CONFIG_IDF_EXPERIMENTAL_FEATURES
  - CONFIG_BOOTLOADER_CACHE_32BIT_ADDR_QUAD_FLASH

  Please note, these options are experimental and may not work stably on all quad flash chips. Additionally, they are only supported on ESP-IDF v5.2 and later.
- The reason for the successful test based on the ESP32-S3-WROOM-2-N32R8 module is that this module defaults to using octal flash. In octal flash mode, the 32 bit Cache function is enabled by default, corresponding to the configuration option `CONFIG_BOOTLOADER_CACHE_32BIT_ADDR_OCTAL_FLASH`.
- For detailed instructions, see Restrictions of 32-bit address support for QSPI flash chips.

When using an XMC-D flash on ESP32-C3 with the flash suspend feature enabled, OTA upgrade fails with the error `only xmc is supported`. How can this be resolved?

The issue has been fixed in ESP-IDF v5.3.3 and later versions. Users on earlier versions need to upgrade to v5.3.3 or later, or contact Espressif for customized support. The root cause is that after enabling flash suspend, spi0 enters the suspend state, but spi1 is not correctly configured, resulting in an error when writing OTA to flash.

4.8.6 SPIFFS Filesystem

How do I store the keys and certs of ESP32 devices in SPIFFS?

You can generate an SPIFFS image from files and flash it to the corresponding partition. See SPIFFS Filesystem for details.

Can ESP32 mount a SPIFFS file system partition in the external SPI flash ?

Yes, this function has been added in ESP-IDF v4.0 and later versions. Please note that when two partitions are mounted to ESP32, it is not permitted for multiple tasks to write files into the same partition at the same time.

4.8.7 Other Storages

Can ESP32 use LittleFS file system?

LittleFS is a third-party component esp_littlefs, which can be used directly in ESP-IDF. You can use the mklittlefs tool for the image of LittleFS file system.

How to check the memory usage (e.g., DRAM, IRAM, rodata) of ESP32 chips?

You can check the estimated occupied static storage of ESP32 chips by inputting the instruction `idf.py size-components` under corresponding directories in terminal. You can use heap_caps_get_info to obtain dynamic applied memory during operation.

What is the available size of RTC RAM in ESP8266 for users?

- The available RTC RAM in ESP8266 for users is 512 bytes (0x200). Please see descriptions in`esp8266.ld <https://github.com/espressif/ESP8266_RTOS_SDK/blob/release/v3.4/components/esp8266/ld/esp8266.ld>`_.

How to enable exFAT?

CHIP: ESP32

- please modify #define FF_FS_EXFAT 0 as #define FF_FS_EXFAT 1 , please refer to ffconf.h for details.

Is there a limit to the number of partitions in the partition table of ESP32?

- Yes. The length of partition table is 0xC00 bytes (can store up to 95 partition table entries). Please refer to the description in partition table.

How to read the remaining memory of the ESP32 chip?

- The remaining memory of the chip RAM can be read through esp_get_free_heap_size().

What should I pay attention to when using `xTaskCreateStatic()` in ESP-IDF?

- You can refer to the xTaskCreateStatic() description.

Based on the ESP-IDF SDK, when defining a global variable with *RTC_NOINIT_ATTR uint32_t counter = 0;*, the printed value is *1368610594*. What is the reason for this?

- The RTC_NOINIT_ATTR variable is compiled into the .rtc_noinit section, which is not cleared at startup. Therefore, assigning a value at variable initialization will not take effect. It only becomes effective when assigned after initialization.
- When using *RTC_NOINIT_ATTR* to define a global variable, it is necessary to assign a value to the variable in the application code. The correct approach is as follows:

```
RTC_NOINIT_ATTR uint32_t counter;
counter = 0;
printf("Counter value: %"PRIu32" \n",counter);
```

4.9 System

4.9.1 How do I disable the watchdog if my application doesn't need it?

There are mainly two types of watchdogs in ESP-IDF: task watchdog and interrupt watchdog. You can disable both types of watchdogs in menuconfig. For more details, please refer to Watchdogs.

4.9.2 What are the differences between RTOS SDK and Non-OS SDK?

The main differences are as follows:

Non-OS SDK

- Non-OS SDK mainly uses timers and callback functions to implement nesting of various functional events, triggering specific functions under certain conditions. Non-OS SDK uses the espconn interface for network operations, and users need to develop software according to the usage rules of the espconn interface.

RTOS SDK

- RTOS SDK is based on FreeRTOS, a multi-tasking OS. You can use the standard FreeRTOS interfaces to implement resource management, loop operations, execution delay, inter-task messaging and synchronization, and other task-oriented process design approaches. For specific interface usage methods, refer to the usage instructions on the official FreeRTOS website or the book USING THE FREERTOS REAL TIME KERNEL - A Practical Guide.

- The network operation interface of RTOS SDK is the standard lwIP API. RTOS SDK provides a package which enables BSD Socket API interface. You can directly use the socket API to develop software applications, and port other applications from other platforms using socket API, effectively reducing the learning and development cost arising from platform switch.
- RTOS SDK introduces the cJSON library whose functions make it easier to parse JSON packets.
- RTOS is compatible with Non-OS SDK in Wi-Fi interfaces, SmartConfig interfaces, Sniffer related interfaces, system interfaces, timer interfaces, FOTA interfaces and peripheral driver interfaces, but does not support the AT implementation.

4.9.3 Why does the log output ets_main.c when ESP8266 starts?

If ESP8266 prints `ets_main.c` when it starts, it indicates there are no programs that can be operated. Please check the binary file and burn address if you encounter this issue.

4.9.4 Why do I get compile errors when using IRAM_ATTR in Non-OS SDK?

If you need to execute a function in IRAM, you don't need to add the `ICACHE_FLASH_ATTR` macro, then this function is executed in IRAM.

4.9.5 Where is the main function of ESP8266?

ESP8266 SDK does not provide the main function. The main function is stored in the first-stage bootloader in ROM, which is used to load the second-stage bootloader. The entry function of the second-stage bootloader is ets_main. After startup, the user_init in the user application will be loaded to lead the user application.

4.9.6 Which part of ESP8266 partition-tables should be paid special attention to?

Compared to those of ESP32, partition-tables of ESP8266 have some special requirements on OTA partitions due to cache characteristics of ESP8266. For details, please refer to Offset & Size of ESP8266 Partition Tables。

4.9.7 Is it possible to compile the binaries in application layer and bottom layer separately?

No, they cannot be compiled separately.

4.9.8 What should I pay attention to when using flash of ESP32 module at 80 MHz?

Espressif modules have undergone stability tests before they are sold, and they support the frequency of 80 MHz. According to stability test data, operating flash at the frequency of 80 MHz will not affect the service life and stability.

4.9.9 How can I download programs into a single-core module using ESP-IDF?

When compiling the program, use the `make menuconfig` command to enter the configuration interface, and make the following configuration to download the program on a single-core module. Please note that in the configuration interface, press Y to enable, and N to disable.

`Component config > FreeRTOS > Run FreeRTOS only on first core`

4.9.10 How to enable the dual-core mode of ESP32 with ESP-IDF?

ESP-IDF is generally configured in dual-core mode by default. You can switch between the single and dual cores in menuconfig: `menuconfig > Component config > FreeRTOS > Run FreeRTOS only on first core`. ESP32 is in the single-core mode if this option is enabled, and is in the dual-core mode if it is disabled (default).

4.9.11 Can the ESP32-D0WD chip store user programs?

No, user programs must be stored in the external flash. The on-chip ROM cannot store user programs. The program stored in the ROM is the chip's first-stage bootloader. To protect the factory program from being damaged, this area is read-only.

4.9.12 Will the data in PSRAM be lost when ESP32 enters low power mode?

- In Modem-sleep/Light-sleep mode, the data in PSRAM will not be lost.
- In Deep-sleep mode, the CPU and most peripherals will lose power, and the data in PSRAM will be lost.

4.9.13 Is the ESP32 CPU system time generated by the system tick clock? What is the precision?

The CPU system time is generated by the internal 64-bit hardware timer CONFIG_ESP_TIMER_IMPL of esp_timer, which has a microsecond-level time resolution. For details, please see High Precision Clock Description.

4.9.14 How can I modify the clock frequency of ESP32's flash and PSRAM?

Modify it in menuconfig:

- flash clock frequency: `menuconfig > Serial flasher config > Flash SPI speed`.
- PSRAM clock frequency: `Component config > ESP32-specific > SPI RAM config > Set RAM clock speed`.

4.9.15 When using ESP32-SOLO-1 modules, what settings should I make to run ESP-IDF on a single-core module?

First, you need to use the `menuconfig` command to enter the configuration interface, and then enable the option `Component config > FreeRTOS > Run FreeRTOS only on first core` to run ESP-IDF on a single-core module.

4.9.16 Can time_t be configured to be 64 bits in ESP-IDF? (It is currently 32 bits)

ESP-IDF uses a 64-bit signed integer to represent time_t starting from release v5.0. For details, please see Unix Time 2038 Overflow.

4.9.17 How does the firmware distinguish whether the main chip is ESP8285 or ESP8266?

External tools such as esptool are typically used to read the chip type. You can read the corresponding register bits of the chip in the firmware according to the Python code example, and then calculate to determine the result.

```
def get_efuses(self):
# Return the 128 bits of ESP8266 efuse as a single Python integer
return (self.read_reg(0x3ff0005c) << 96 | self.read_reg(0x3ff00058) << 64␣
↪| self.read_reg(0x3ff00054) << 32 | self.read_reg(0x3ff00050))

def get_chip_description(self):
  efuses = self.get_efuses()
  is_8285 = (efuses & ((1 << 4) | 1 << 80)) != 0  # One or the other␣
↪efuse bit is set for ESP8285
  return "ESP8285" if is_8285 else "ESP8266EX"
```

4.9.18 Can ESP32 load library files as dynamic libraries?

ESP32 does not support loading library files as dynamic libraries. It only supports static libraries.

4.9.19 How can I reduce the IRAM occupied by the ESP32 system?

- Please disable `menuconfig > Component config > LWIP > Enable LWIP IRAM optimization` by typing `N`.
- Please change the configurations in `menuconfig > Compiler option > Optimization Level > Optimize for size (-Os)`.
- Please disable `WiFi IRAM speed optimization (N)` and `WiFi RX IRAM speed optimization (N)` in `menuconfig > Component config > wifi`.
- For more details, please refer to Minimizing RAM Usage。

4.9.20 What is the low voltage reset threshold of the ESP32 chip?

The brownout voltage threshold ranges from 2.43 V to 2.80 V, which can be set in `menuconfig > Component config > ESP32-specific > Brownout voltage level`.

4.9.21 Why is ESP32 automatically woken up in the light sleep example?

In the light sleep example, two wake-up methods are used by default as follows:

```
esp_sleep_enable_timer_wakeup(2000000);      // Automatically wake up␣
↪after 2 seconds
esp_sleep_enable_gpio_wakeup();              // GPIO wake-up
```

By default, the GPIO wake-up method wakes ESP32 up when GPIO0 is in the low level. When GPIO0 is low, ESP32 is in the wake-up state. When GPIO0 is high, ESP32 automatically enters the Light-sleep mode. If you need to maintain the Light-sleep mode for a long time, you can command out automatic wake-up after 2 seconds and only enable GPIO wake-up.

4.9.22 When testing ESP32 deep_sleep example, why does the program crash into a dead loop when const int wakeup_time_sec = 3600?

The program crashes because the int type parameter *wakeup_time_sec* overflows during the operation of wakeup_time_sec * 1000000.

```
const uint64_t wakeup_time_sec = 3600;
printf("Enabling timer wakeup, %lldn",wakeuo_time_sec);
```

4.9.23 How many system reset methods does ESP32 have?

ESP32 has several system reset methods, including the following:

- Software Reset: In the application program, you can perform a software reset by calling the esp_restart() function.
- External Reset: ESP32 can be reset by external hardware circuits, such as pressing the RESET button, unstable power supply voltage, etc.
- Hardware Watchdog Reset: When ESP32 encounters a deadlock or other abnormal conditions during operation, the hardware watchdog module will automatically trigger a reset.
- Brownout Reset: When the system voltage is unstable or the power supply voltage is relatively low, ESP32 ' s built-in power management module will automatically trigger a reset.
- Exception Reset: When ESP32 encounters a CPU exception during operation, such as accessing illegal memory, running illegal instructions, etc., it will trigger an exception reset.
- JTAG Reset: When debugging ESP32 with a JTAG debugger, you can reset it through the JTAG reset signal.
- For more details, see Section 4.1.2 Reset Sources in ESP32 Technical Reference Manual.

4.9.24 Why does ESP8266-NONOS-V3.0 SDK output the following error?

```
E:M 536
E:M 1528
```

- The log starting with E:M is caused by insufficient remaining memory.

4.9.25 Can ESP32 use the entire 8 MB of PSRAM memory?

- Yes.
- Since the maximum cache mapping space is 4 MB, only 4 MB of PSRAM can be used by mapping, and the remaining space can be used through API.
- For details, please refer to the example himem.

4.9.26 After ESP8266 AT connects to AP, the system enters into Modem-sleep by default, but the current does not drop significantly. What are the reasons?

- After the AT firmware connects to the AP, ESP8266 will enter into Modem-sleep mode automatically, and the power consumption will fluctuate between 15 mA ~ 70 mA.
- If the power consumption does not fluctuate between 15 mA ~ 70 mA, and the current does not show a waveform in the oscilloscope, please refer to the following suggestions: - Erase the device flash and re-flash the AT firmware. - Capture network packets to analyze whether there are devices that frequently send broadcast packets in the current network environment. If yes, you can test with a router (AP) in another network environment.

4.9.27 Can ESP32 permanently change the MAC address?

- The MAC address that comes with the chip cannot be modified. The eFuse supports users to write their own MAC address.
- The customized MAC address can be obtained by calling API in the firmware, and it can be set to replace the default address in the system.
- For detailed configuration, please refer to mac-address.
- In addition, Espressif provides a service to burn the MAC address provided by the user before the chip leaves the factory. If you have such requirements, please email sales@espressif.com.

4.9.28 How can ESP8266 verify all.bin as an illegal file during OTA upgrade?

Background:

- all.bin is generated by merging bootloader.bin, partition.bin, and app.bin.
- ota.bin is the object bin file used for OTA upgrade.

When using simple_ota_example for OTA upgrade, if you mistakenly download all.bin from the server and write it into the OTA partition, the device will repeatedly restart.

Cause analysis:

The code does not verify all.bin, resulting in writing illegal bin files into the OTA partition.

Solution:

all.bin can be judged as an illegal bin file by turning on sha256 verification. The configuration is as follows: `Component config > App update > [*] Check APP binary data hash after downloading`.

4.9.29 Where are the release notes after the ESP-IDF version is updated?

For the release notes, please see GitHub release note.

4.9.30 Does ESP8266 have a detailed register manual?

Please refer to ESP8266 TRM > Appendix.

4.9.31 ESP32 cannot start normally after enabling Secure Boot and outputs the following error. What is the reason for it?

```
csum err:0x9a!=0x5f
ets_main.c 371
ets Jun  8 2016 00:22:57
rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:4
load:0x3fff0034,len:9372
load:0x40078000,len:19636
```

This may be because the Bootloader become larger after enabling Secure Boot, and the bin file is overwritten when the firmware is burned. You can check the size of the Bootloader after Secure Boot. For example, you can try to increase the offset of the partition table to 0xF000.

4.9.32 How does ESP8266 save data during a software reboot?

- If writing or modifying operations are not frequent, flash can be used to store data. This area is relatively larger than memory and easy to be adjusted.
- If the data is small, RTC Memory can be used to store related data. Please see the interface in esp_system.h in the branch of Rel 2.1 (refer to usage instructions) system_rtc_mem_read.
- If neither of the above can meet the requirements, you can also choose the external RTC memory, which can interact with I2C and SPI.
- It is recommended to write the data to flash when the writing frequency is not high, because the data will not lose when the power is off.

4.9.33 What timers are available on ESP8266?

- ESP8266 has a hardware timer that can generate interrupts. Calling API in NONOS SDK and RTOS SDK are slightly different.
- Software timer:
  - The API os_timer in NONOS is a DSR process, which cannot generate interrupts, but can generate tasks. Tasks will be queued according to the ordinary level.
  - In RTOS, you can use the software timer in FreeRTOS, which is more flexible.

4.9.34 What is the purpose of the watchdog on ESP8266?

- In order to provide system stability and cope with the operating environment with multiple conflicts, ESP8266 integrates a 2-level watchdog mechanism, including software watchdog and hardware watchdog.
- Both watchdogs are enabled by default. HW WDT is always running. If the HW WDT is not reset, the MCU will be reset after about 6 seconds.
- SW WDT will reset the MCU in about 1.5 seconds. You can enable/disable SW WDT, but you cannot enable/disable HW WDT. Because you must reset the SW WDT before you can reset the HW WDT at the same time.
- The watchdog can be configured by modifying `make menuconfig` > `Component config` > `Common ESP-related` > `Invoke panic handler on Task Watchdog timeout`.

4.9.35 What should be paid attention to when using `user_init` in ESP8266?

- `wifi_set_ip_info` and `wifi_set_macaddr` only take effect when called in `user_init`. They do not take effect when called elsewhere.
- It is recommended to call `system_timer_reinit` in `user_init`. Otherwise, you need to re-arm all timers after calling it.
- If `wifi_station_set_config` is called in `user_init`, the underlying layer will automatically connect to the corresponding router, and there is no need to call `wifi_station_connect` to connect. Otherwise, you need to call `wifi_station_connect` to connect.
- `wifi_station_set_auto_connect` sets whether to automatically connect to the recorded router when powered on. For example, if the automatic connection function is turned off, and it is called in `user_init`, then the router will not be automatically connected when powered on this time. If it is called elsewhere, the router will not be automatically connected when powered on next time.

4.9.36 Why does the system keep restarting after ESP32 enables both `Enable debug tracing of PM using GPIOs` and `Allow .bss segment placed in external memory`?

- The `Enable debug tracing of PM using GPIOs` configuration option needs to be enabled during GDB debugging and cannot be used simultaneously with the `Allow .bss segment placed in external memory` configuration option.
- Because `Enable debug tracing of PM using GPIOs` uses GPIO16 and GPIO17 by default, which conflicts with the PSRAM interface (also default to GPIO16 and GPIO17).

4.9.37 Why does the program trigger RTCWDT_RTC_RESET when ESP32 bootloader in ESP-IDF v3.3 version runs app bin in v3.1 version?

- The bootloader v3.3 enables the WDT watchdog and turns off the WDT watchdog when the application (app) is running.
- However, bootloader v3.1 does not enable the WDT watchdog, so the application (app) does not have the WDT watchdog mechanism, which leads to bootloader v3.3 triggering the WDT watchdog reset when guiding the application (app) v3.1.
- You can disable `BOOTLOADER_WDT_ENABLE` in `menuconfig` to turn off the WDT watchdog in bootloader v3.3.

4.9.38 Does the ESP32 chip come with a unique chip_id from the factory?

The ESP32 chip does not have a unique chip_id, but it has a globally unique MAC address by default, which can be used to replace the chip_id.

4.9.39 How to check ESP8266 reset cause?

Please refer to ESP8266 Exception Restart Causes.

4.9.40 How can I optimize the size of binary files compiled by ESP32?

The bin file compiled for ESP32 usually includes application code, partition table, ESP-IDF firmware, and other data. To optimize the size of it, the following methods can be adopted:

- Configure compilation options: GCC compilation can be optimized by configuring `idf.py menuconfig` > `Compiler options` > `Optimization level (Optimize for size(-Os))`.
- Optimize code: The application code can be optimized, such as adopting more efficient algorithms and data structures, simplifying code logic and process, improving code reuse rate, adjusting log level, reducing unnecessary log printing, and reducing code file size.
- It should be noted that when optimizing the size of the bin file, the optimization effect and program function need to be balanced to avoid excessive optimization causing program exceptions or incomplete functions. It is recommended to refer to the official documents and examples when optimizing the size of the bin file, and follow the relevant regulations and standards.

For more details, please refer to Minimizing Binary Size.

4.9.41 Does ESP32 have an API for rebooting the system?

- You can use the API `esp_restart()` to reboot the system. For related instructions, please refer to documentation.

4.9.42 What is the reason for the exception log `invalid header: 0xffffffff` of ESP32?

The ESP32 chip usually prints this exception log in the following situations:

- The power-on and power-off timing of the chip is incorrect, and some areas of the chip are not fully reset.
- There is an exception in the firmware in flash, such as incomplete firmware burning.
- The flash device is damaged and cannot read the correct data.
- The chip's own cache is turned off or damaged and cannot read firmware data.

4.9.43 What is the timed mechanism to wake up ESP8266 from deep sleep?

In Deep-sleep mode, connect GPIO16 (XPD_DCDC) to EXT_RSTB. After the sleep time is reached, GPIO16 outputs a low level to the EXT_RSTB pin, and the chip is reset and awakened.

4.9.44 Why is the ESP32 RAM obtained using `heap_caps_get_free_size` about 300 KB, instead of the 520 KB provided in the manual?

- This is because the memory is pre-allocated to various function modules at system startup, and the remaining memory after system startup is about 300 KB.
- If the remaining memory is insufficient, you can choose a module with PSRAM, and allocate the memory in the PSRAM.

4.9.45 How to perform OTA upgrade via LAN app for ESP32 & ESP8266?

- Devices within the LAN can configure to enable the HTTP service, and send the provided firmware download link to the device through other methods (UDP, CoAP, MQTT, etc.).
- The device can complete the OTA update through the traditional URL OTA method, and the application example is provided in the SDK.

4.9.46 How can I modify the GPIO used by the serial port for log output in ESP32?

- Configure `menuconfig` > `Component Config` > `ESP System Settings` > `Channel for console output` > `Custom UART`, and select custom UART pin.
- Go back to the previous level, you will see the options of `UART TX on GPIO#` and `UART RX on GPIO#`. By modifying these two options, you can change the GPIO used by the serial port for log output.

4.9.47 ESP8266 uses MQTT ssl_mutual_auth to communicate. Why does the following error occur during OTA?

- The error 0x7f00 is due to insufficient memory. It is recommended to use HTTP for OTA.

4.9.48 There are NVS options in ESP32's `menuconfig` > `Component config`. Why are the configuration items are empty?

- The NVS option in `menuconfig` > `Component config` is to configure the NVS encryption function, and the prerequisite for this function is to enable flash encryption.
- After configuring the option `menuconfig` > `security features` > `enable flash encryption on boot option`, you can see the NVS configuration options.

4.9.49 Does ESP32 undergo a random watchdog reset after power-up or waking up from Deep-sleep?

- The watchdog reset on power-up cannot be bypassed by software, but ESP32 starts normally after reset.
- The watchdog reset after waking up from Deep-sleep is automatically bypassed in ESP-IDF v1.0 and higher versions.
- After waking up from Deep-sleep, the CPU can immediately execute a program in RTC fast memory. The program in RTC fast memory bypasses the watchdog reset after waking up from Deep-sleep by clearing the illegal access flag of cache MMU. Specifically, it
  - Sets the `PRO_CACHE_MMU_IA_CLR` bit of the `DPORT_PRO_CACHE_CTRL1_REG` register to 1.
  - Clears this bit.

4.9.50 When the ESP32 CPU uses cache to access external SRAM, if these operations need to be processed by the CPU at the same time, will read and write errors occur?

- This problem cannot be automatically bypassed by software.
- For ESP32 version 0, when CPU uses cache to access external SRAM, it can only perform unidirectional operations, that is, it can only perform write operations or read operations to SRAM. These two operations cannot be realized alternatively.
- Use the MEMW instruction: after the read operation, add the `__asm__("MEMW")` instruction, and then initiate the write operation before the CPU pipeline is cleared.

4.9.51 ESP32 CPU may freeze when it switches directly from 240 MHz to 80/160 MHz. How can I solve it?

- It is recommended to use the following two modes:
  (1) 2 MHz <-> 40 MHz <-> 80 MHz <-> 160 MHz
  (2) 2 MHz <->40 MHz <->240 MHz
- This issue has been fixed in chip version 1.

4.9.52 For ESP32 pads with both GPIO and RTC_GPIO functions, the pull-up and pull-down resistors can only be controlled by the RTC_GPIO pull-up and pull-down registers. How can I solve it?

- The GPIO driver in ESP-IDF v2.1 and higher versions automatically bypasses this issue.
- Both GPIO and RTC_GPIO use the RTC_GPIO register.

4.9.53 As the flash startup speed is slower than that of the chip reading flash, ESP32 may randomly reset the watchdog once after power-on or waking up from Deep-sleep. How can I solve it?

- Use a faster flash instead, requiring the time from flash power-on to being readable to be less than 800 µs. This method can bypass the watchdog reset when the chip is powered on and wakes up from Deep-sleep.
- The issue of watchdog reset after waking up from Deep-sleep is automatically bypassed in ESP-IDF v2.0 and higher versions (the delay time can be configured as needed). Specifically, CPU first reads the instructions in the RTC fast memory after waking up from Deep-sleep, and then reads the flash after waiting for a while.

4.9.54 When the ESP32 CPU accesses the external SRAM, there is a small probability of causing reading and writing errors. How can I solve it?

- When x>=y, insert 4 nop instructions between store.x and load.y.
- When x<y, insert a memw instruction between store.x and load.y.

4.9.55 For a dual-core ESP32 chip, when one CPU's bus is reading address space A, and the other CPU's bus is reading address space B, the CPU reading address space B may encounter an error. How can I solve it?

- When one CPU is reading address space A, avoid another CPU initiating a read operation on address space B through locking and interrupting.
- Before one CPU reads address space A, add an operation of this CPU reading address space B (non-FIFO address space, such as 0x3FF40078), and ensure that the operations of reading address space B and reading address space A are atomic.

4.9.56 The ESP32 CPU resets the interrupt signal of the CAN controller by reading the `INTERRUPT_REG` register. If the CAN controller happens to generate a transmission interrupt signal within the same APB clock cycle, the transmission interrupt signal is lost. How can I solve it?

From the transmission request is initiated till the data transmission completes, each time after reading `INTERRUPT_REG`, you should check the `STATUS_TRANSMIT_BUFFER` bit. If the bit is set and `CAN_TRANSMIT_INT_ST` is not set, the transmission interrupt signal will be lost. In ESP32, the interrupt signal of the CAN controller can be reset by reading the `INTERRUPT_REG` register. However, if the CAN controller generates a transmission interrupt signal within the same APB clock cycle, the interrupt signal may be lost because ESP32 may have cleared it when reading the register within this clock cycle. To solve this problem, the following methods can be used:

- Add delay: Before reading the `INTERRUPT_REG` register, a certain delay can be added to ensure that the interrupt signal of the CAN controller has been cleared. The appropriate delay can be determined through test and adjustment.
- Use interrupt handler: An interrupt handler can be used to handle the interrupt signal of the CAN controller and avoid reading the `INTERRUPT_REG` register within the same APB clock cycle. The interrupt handler can respond to the interrupt signal of the CAN controller in time to ensure that the signal will not be lost.
- Use other registers: Other registers can be used to reset the interrupt signal of the CAN controller to avoid reading the `INTERRUPT_REG` register within the same APB clock cycle. For example, the `CANCTRL` register or `ERRCNT` register can be used.

It should be noted that when using the above methods, it is necessary to choose and implement the appropriate one according to the specific application scenario and requirements. At the same time, sufficient testing and verification of software and hardware are also required to ensure the reliability and stability of the system. When resetting the interrupt signal of the CAN controller in ESP32, it is necessary to avoid losing the interrupt signal to ensure the normal operation of the system.

4.9.57 When the program meets the following conditions simultaneously, ESP32 v3.0 will be live-locked, causing the CPU to remain in a memory access state and unable to continue executing instructions. How can this be resolved?

Please refer to ESP32 Series SoC Errata > subsection 3.15.

4.9.58 The ESP32 CPU has restrictions when accessing the address spaces `0x3FF0_0000 ~ 0x3FF1_EFFF` and `0x3FF4_0000 ~ 0x3FF7_FFFF`. How can this be resolved?

Please refer to ESP32 Series SoC Errata > subsection 3.16.

4.9.59 How can I turn off log output in ESP32 ?

- You can turn off the bootloader log by setting `menuconfig > bootloader config > bootloader log verbosity` to `No output`.
- You can turn off the program log by setting `menuconfig > Component config > log output > Default log verbosity` to `No output`.
- For ESP-IDF release/v4.3 and earlier versions, you can turn off UART0 output log by `menuconfig > Component Config > Common ESP-related > Channel for console output > None`.
- For ESP-IDF release/v4.4 and later versions, you can turn off UART0 output log by `Component config > ESP System Settings > Channel for console output > None`.

4.9.60 Can the data stored in RTC Memory run when ESP8266 is in Deep-sleep mode?

When ESP8266 is in Deep-sleep mode, only the RTC timer continues to work. The data saved in the RTC Memory will not run, but can still be saved here. However, the data saved in RTC memory will lose after ESP8266 is powered off.

4.9.61 What is the maximum length of the NVS Key for ESP32?

- The maximum length of the NVS key for ESP32 is 15 characters, which cannot be changed. Please see the description of key-value pair.
- But you can use the value of nvs_set_str() to store data.

4.9.62 Does cJSON in ESP-IDF release/v4.2 support uint64_t data parsing?

No. The cJSON library has restrictions on parsing long integers, and the longest type it can parse is the Double type.

4.9.63 Given that the GDB debugging function is working before the flash encryption is disabled, then why does the device keep restarting during the GDB debugging after the flash encryption is enabled?

The JTAG debugging function will be disabled by default when flash encryption or secure boot is enabled. For more information, please refer to JTAG with Flash Encryption or Secure Boot.

4.9.64 When ESP32 uses mobile hotspot for OTA firmware download, if the data switch is turned off for a few seconds and then turned on again, the program will always be stuck in OTA (the same applies when unplugging and plugging the WAN cable when using a router). Why?

- This is a normal situation based on the protocol. When using the `esp_https_ota` component to run OTA, you can set the network timeout value `http_config->timeout_ms` to 10 ~ 30 seconds (not recommended to be too small), and enable `http_config->keep_alive_enable` to see if there are any errors at the link layer.
- If you are using a self-implemented OTA module, please set a timeout value via the `select` configuration or enable the TCP keep-alive mechanism to detect the link layer.

4.9.65 Which GPIOs can be used to wake up ESP32-C3 in Deep-sleep mode?

Only GPIO0 ~ GPIO5 in VDD3P3_RTC domain can be used to wake up ESP32-C3 from Deep-sleep mode. Please read Chapter 5.9.1 Power Supplies of GPIO Pins in ESP32-C3 Technical Reference Manual.

4.9.66 When using the ESP-WROOM-02D module with a battery for power supply, are there any risks in frequently formatted reading and writing flash as the battery is low (the module barely starts up)?

Frequent formatting and read/write operations on flash in low power situations may have some risks. It may not work properly or be susceptible to cause errors under low power conditions. In addition, frequent formatting and read/write operations on flash in this situation may lead to the following risks:

- Data loss or corruption: flash may not be able to write data properly under low power conditions. Frequent formatting and read/write operations may result in data loss or corruption.
- Module crash or damage: Frequent formatting and read/write operations on flash in low power conditions will consume the module's power, which may cause the module to crash or damage.

Therefore, it is recommended to minimize access and operations on flash in low power conditions and avoid frequent formatting and read/write operations. If formatting and read/write operations are necessary, ensure that the module has sufficient power, backup data before the operation to prevent data loss, use low power mode and optimize code to minimize power consumption.

4.9.67 How can I check the maximum stack size used by a thread for ESP32?

You can call the UBaseType_t uxTaskGetStackHighWaterMark(TaskHandle_t xTask) function to check it. This function will return the minimum remaining stack space after the task is started.

4.9.68 Why does ESP32 print the "SW_CPU_RESET" log?

On ESP32, printing the "SW_CPU_RESET" log is usually caused by the program terminating abnormally. ESP32 has two built-in processor cores, namely the main core and the auxiliary core. In some cases, if the program is executed on the main core and some abnormal situations occur, such as accessing illegal addresses or unhandled interrupts, it may cause the main core to enter an abnormal state and restart. When this happens, ESP32 will print the "SW_CPU_RESET" log on the serial

terminal (UART). In addition, when developing applications using ESP-IDF, you may also call the `esp_restart()` function in the application program to restart ESP32. In this case, ESP32 will also print the "SW_CPU_RESET" log on the serial terminal. It should be noted that the appearance of the "SW_CPU_RESET" log does not necessarily mean that there is a problem with the program or ESP32 hardware. It may just be a normal phenomenon caused by some abnormal situations. However, if the program frequently encounters exceptions and restarts, debugging and troubleshooting are needed. The cause of the problem can be determined by checking the program log and the status of the hardware device.

4.9.69 For ESP32 products, when testing NVS separately, I found it occupies a lot of memory. What is the reason?

Please check the partition table settings. It is recommended to set a smaller NVS data partition in the partition table to test. The larger the NVS data partition setting, the more memory it will occupy.

4.9.70 How do I change the system time of a module?

CHIP: ESP32 | ESP32 | ESP32-C3

- You can use the `time()` interface in C library to set the system time.

4.9.71 During the OTA upgrade process, an `ESP_ERR_OTA_VALIDATE_FAILED` error occurred after calling `esp_ota_end`. How can I troubleshoot such issue?

CHIP: ESP32

- Generally it is caused by the error content in the downloaded firmware. You can dump out such content via read_flash in esptool from your module. Then use the Beyond Compare tool to compare the two bin files in hexadecimal to see which part of the bin file is downloaded incorrectly.

4.9.72 How does ESP8266-RTOS-SDK store data to RTC memory?

- The definition method of storing data in RTC memory is as follows:

```
#define RTC_DATA_ATTR _SECTION_ATTR_IMPL(".rtc.data", __COUNTER__)
```

- Please refer to the description in esp_attr.h.

4.9.73 After waking up from Deep-sleep mode, where does ESP8266 boot?

After ESP8266 wakes up from Deep-sleep mode, the device will boot up from `user_init`. Please refer to the description in esp_deep_sleep().

4.9.74 When will the RTC clock be reset?

Any reset (except the power-up reset) or sleep mode settings will not reset the RTC clock.

4.9.75 Can ESP32 be woken up by pulling EN low after entering Deep-sleep mode with the `AT+GSLP` command?

- Yes, but it is not recommended.
- Waking up from Deep-sleep mode can be realized by RTC_GPIO. Please refer to ESP32 Technical Reference Manual.

4.9.76 When multiple threads want to use the watchdog of ESP32, should each thread enable the watchdog individually?

Yes, please see Task watchdog instructions.

4.9.77 How to enter Light-sleep mode with ESP8266-RTOS-SDK release/v3.3?

- First set the wake-up mode of Light-sleep mode, please refer to ESP8266_RTOS_SDK/components/esp8266/include/esp_sleep.h.
- Then use the esp_light_sleep_start() API to enter Light-sleep mode.
- For the program implementation logic, please refer to esp-idf/examples/system/light_sleep/main/light_sleep_example_main.c.
- For API descriptions about sleep modes in ESP8266-RTOS-SDK, please see Sleep modes API Reference.

4.9.78 How can I wake up ESP8266 from Deep-sleep mode?

ESP8266 can only be awakened from Deep-sleep mode via RTC Timer. The timing duration is set by user via `esp_deep_sleep()`, and GPIO16 (XPD_DCDC) should be connected to EXT_RSTB through a 0 Ω resistor to support such function. Please refer to related API descriptions.

4.9.79 When using the ESP32-WROVER* module, there is a problem of battery jitter or abnormal power-off and power-on, causing the system to crash and fail to wake up. What is the reason?

- Application scenario: When sleeping, the current is about 12 uA. When the battery is unplugged or the product is shaken, it will cause power failure, but there is still electricity in the capacitor. The process of ESP32 discharging from 3.3 V to 0 V, and then powering on to restore 3.3 V will cause ESP32 to be unable to wake up.
- Please check whether the chip VCC and EN meet the power-on timing requirements.
- When using the ESP32-WROVER* module for sleep, if there is unstable power supply voltage or abnormal power off, it may cause problems with the power management unit of the chip, leading to inability to wake up normally.
- Consider adding a reset chip to ensure normal timing.
- For the power-on and reset timing description of ESP32, please refer to the ESP32 Datasheet.

* indicates that the product is in EOL status.

4.9.80 How can I flash a customized MAC address?

You can start by understanding the MAC mechanics of ESP modules, please refer to Introduction to Mac Addresses. There are currently two options for burning customized MAC addresses:

- Option 1: directly flash it into eFuse blk3.
- Option 2: Store in flash. It is not recommended to store the MAC address in the default NVS partition. It is recommended to create a customized NVS partition for storing customized Mac addresses. For more information on customized MAC addresses, please refer to base_mac_address.

4.9.81 When ESP32 uses esp_timer, network communication or Bluetooth communication is abnormal. What is the reason?

- esp_timer is a high-precision hardware timer component, and some background components also use it to complete some system tasks. When using esp_timer, please do not call delay and blocking APIs in the callback function of the timer, and try to ensure that the function is executed as quickly as possible, so as not to affect the performance of other system components.
- If you do not require high timing accuracy, please use the timer component xTimer in FreeRTOS.

4.9.82 With ESP32, are there any return instructions if I skip to a function using the `jump` instruction in ULP ?

For ULP CPU instructions list and corresponding specifications, please see ULP Coprocessor Instruction Set. Normally, a general register is used for return instructions to store backup PC addresses for later jumping backs. Since there are only four general registers in ULP for now, please make proper use of them.

4.9.83 How to adjust the warning level for project build?

When building the project, it is found that some warnings being treated as errors, causing build failure, as follows:

```
error: format '%d' expects argument of type 'int *', but argument 3 has␣
↪type 'uint32_t *' {aka 'long unsigned int *'} [-Werror=format=]
```

For the error above, you can modify compilation flags at a component level (in the CMakeLists.txt file in a component) or at a project level (in the CMakeLists.txt file in a project). These two ways have roughly the same effect.

- To modify compilation flags for a specific component, use the standard CMake function `target_compile_options`. Please refer to Controlling Component Compilation. For an example of `target_compile_options` at the component level, please see CMakeLists.txt#L3.
- To modify compilation flags for the whole project, either use standard CMake function `add_compile_options` or IDF-specific function `idf_build_set_property` to set `COMPILE_OPTIONS` property. Please refer to overriding-default-build-specifications.

4.9.84 The firmware compiled based on the ESP-IDF SDK varies as it contains the information about `IDF_PATH` and compilation time. How to remove that information?

- For SDK v5.0 and the above versions, you can enable the `CONFIG_APP_REPRODUCIBLE_BUILD` configuration option. In doing so, the application built upon ESP-IDF does not depend on the build environment and both the .elf file and .bin file of the application remain unchanged even if the following variables change:
  - Directory where the project is located
  - Directory where ESP-IDF is located (IDF_PATH)
  - Build time

  Please refer to the Reproducible Builds description.
- For SDK versions below v5.0, you can disable `CONFIG_APP_COMPILE_TIME_DATE=n` to remove the built timestamp information and enable `COMPILER_HIDE_PATHS_MACROS=y` to hide `IDF_PATH`.

4.9.85 When I downloaded the official application hello_world using ESP32-S3-DevKitM-1, the following error occurred. What is the reason for that?

```
ESP-ROM:esp32s3-20210327
Build:Mar 27 2021
rst:0x7 (TG0WDT_SYS_RST),boot:0x8 (SPI_FAST_FLASH_BOOT)
Saved PC:0x40043ac8
Invalid chip id. Expected 9 read 4. Bootloader for wrong chip?
ets_main.c 329
```

- The current error may be related to the chip version on the development board or to the fact that the software version of the ESP-IDF SDK is not the official production version. The chip (ROM) bootloader expects the chip ID is 9, which is the production version of the chip (not a test version). However, in the secondary bootloader header, it sees the chip ID is 4, which is the beta version of the chip. Please refer to the description in esp-idf/issues/7960.
- The actual version of the chip can be obtained by the command `esptool.py chip_id`. If the chip version is the production version, this error is related to the version of the used ESP-IDF SDK. For ESP32-S3 series products, ESP-IDF release/v4.4 and later are necessary.

4.9.86 What is the accuracy of the internal 150 kHz RTC of ESP32 series chips?

The accuracy of the internal 150 kHz RTC of ESP32 series chips is ±5%.

4.9.87 What versions of ESP-IDF SDK are supported by ESP32-D0WDR2-V3 chip?

The supported ESP-IDF versions are: v4.4.1, v4.3.3, v4.2.3, v4.1.3.

4.9.88 When testing OTA applications based on the ESP32 chip, can I delete the default factory partition in the partition table and set the address of the OTA_0 partition to 0x10000?

Yes. Please note that the offsets of partitions of any app type have to be aligned to 0x10000 (64K).

4.9.89 Why can't the `espefuse.py burn_key` command be used to burn ESP32-C3 eFuse BLOCK3?

- `espefuse.py burn_key` can only burn data to eFuse blocks of the KEY_DATA type. However, BLOCK3 of ESP32-C3 is of the USR_DATA type by default.
- You can burn data to eFuse blocks of the USR_DATA type with `espefuse.py burn_block_data`.

4.9.90 Why I encountered the following error after I run the firmware based on the ESP-IDF SDK?

```
***ERROR*** A stack overflow in task sys_evt has been detected.
```

The error is caused by insufficient system_event task stack. You can try to resolve it by increasing `Component config` > `ESP System Setting` > `Event loop task stack size`. However, the overflow occurs because too much logic is being processed within system_event. It is not recommended as it might lead to delayed handling of subsequent events. We suggest forwarding this event to other tasks for processing, either through a queue or other operations.

4.9.91 How to solve the issue of being unable to parse due to spaces in the specified url during Wi-Fi OTA?

Spaces can be replaced with + or `%20` to solve this issue.

4.9.92 How to get the version number of newlib of ESP-IDF?

- There are two ways to obtain the version number:
  - 1. Run *xtensa-esp32-elf-gcc -dM -E -x c - <<< "#include <_newlib_version.h>" | grep NEWLIB_VERSION* command to obtain the newlib version number. The printed log should be similar to the following: *#define _NEWLIB_VERSION "4.1.0"* .
  - 2. Search for the newlib version in the toolchain version and find the toolchain version used by ESP-IDF. For example, for ESP-IDF v5.0, you can get the version of the toolchain is esp2021-r1 from xtensa esp32 elf. Go to the release note page of this toolchain version, it can be seen from the link that the newlib version is v4.1.0.

4.9.93 Does ESP32-P4 support floating-point arithmetic?

- The HP CPU of ESP32-P4 supports floating-point arithmetic, but the LP CPU does not.

4.9.94 What are the ESP-IDF versions supported by different series of ESP32 chip versions?

- Please refer to the ESP-IDF version compatibility with Espressif chip versions description.

4.9.95 How to check the descriptions of error codes defined in ESP-IDF?

- See: Error Code Reference.

4.9.96 Why does the following error occur during firmware operation? What could be the common reason?

```
Guru Meditation Error: Core  1 panic'ed (Unhandled debug exception).
Debug exception reason: Stack canary watchpoint triggered (zcr_task)
```

- The aforementioned log error is usually caused by a stack overflow. You can try increasing the stack size of the zcr_task task.
- For more information on software exception explanations, refer to Severe Errors.

4.9.97 Does the ESP chip support using the esp-bootloader-plus compression upgrade scheme to upgrade other MCUs?

No, it's not possible. Compression update for other MCUs is something the user needs to consider. In the compression update of ESP32, the decompression occurs during the bootloader stage, instead of the app stage, so it cannot directly decompress data for other MCUs. However, you can first decompress the data in ESP32 apps, and then send the decompressed data to other MCUs. You can implement this process by yourself. For details, please refer to xz_decompress_file decompression example.

4.9.98 When developing applications based on ESP32, how can I obtain information such as task status, task priority, remaining task stack, and the core used by the task?

- You can use the vTaskList() function based on FreeRTOS to get relevant information.

4.9.99 How to obtain the CPU usage of a task when developing applications with ESP32?

- You can use vTaskGetRunTimeStats() based on FreeRTOS to get the CPU usage of system tasks.

4.9.100 After downloading the firmware, ESP32 fails to start normally, and the log shows the following information. What could be the reason?

```
E(88)flash parts: partition 0 invalid magic number 0x5e9
E(95)boot: Failed to verify partition table
E(100)boot: load partition table error!
```

- The above error is usually caused by a mismatch between the address of the downloaded `partition-table.bin` and the software setting for `Partition Table > Offset of partition table`. In other words, the downloaded address of `partition-table.bin` is incorrect.
- After the project is built, a `build` folder is created. Insider the `build` folder, there is a `flash_project_args` file, which stores the `bin` file generated by the project compilation and the corresponding download address information.

4.9.101 Does it support redirecting the UART0 output logs of ESP32 to the file system?

Yes. UART0 output logs can be redirected to a file through the esp_log_set_vprintf API.

4.9.102 Can the ESP32 BootLoader, which is configured to run in single-core mode, be upgraded to dual-core mode via OTA?

- ESP32 does not support this. Each CPU in ESP32 has an independent cache, and the MMU configuration related to the cache is set during the BootLoader. If the BootLoader is configured for single-core mode, but the MMU for the second core is not configured, it will cause an instruction fetch error.
- Support is available for ESP32-S3 and ESP32-P4. These two chips feature two cores that share the same cache, which eliminates the aforementioned issue. Therefore, they support upgrading from single-core to dual-core mode.

4.9.103 Does the ESP32 series chip support OTA firmware updates via File Transfer Protocol (FTP)?

The default example does not support TLS-based FTP and only implements basic socket programming. However, ESP32 series chips do support TLS-based FTP, and users can implement this feature if needed.

4.9.104 After using the native_ota_example for an OTA upgrade, why does the device enter the ESP_OTA_IMG_UNDEFINED event, with the ota_state value printed as -1, even though the firmware works normally?

This may be due to the application rollback feature not being enabled in menuconfig. Please ensure that the application rollback feature is enabled in the configuration to avoid this issue.

4.9.105 What are the differences between sdkconfig, sdkconfig.default, and sdkconfig.old in an ESP-IDF project?

- sdkconfig.default is the default configuration file for the project. If there is no sdkconfig file under the project, a sdkconfig file will be generated based on the sdkconfig.default file configuration during compilation; if a sdkconfig file already exists, the configuration from that file will be used directly during compilation.
- The sdkconfig.old file is a backup file automatically generated when recompiling after modifying the configuration options in the sdkconfig file. After each modification and recompilation, the new configuration is saved in the sdkconfig file, while the previous configuration is preserved in the sdkconfig.old file.
- When modifying configuration options through the menuconfig command, the changes are actually applied to the sdkconfig file.
- The firmware configuration options after project compilation depend on the settings in the sdkconfig file.

4.9.106 What could possibly be the reason for the "Not digestSign key in json doc of OTA" error that occurs during an OTA upgrade?

This error message indicates that the security OTA is not enabled, but it does not affect the upgrade process. If you are using a non-secure OTA, this error can be ignored. If security OTA is required, it is recommended to check the platform configuration to ensure that the security OTA feature is enabled.

4.9.107 Are there any firmware update solutions that can help reduce flash memory and transmission traffic?

- The bootloader_support_plus component provides compressed update functionality, which can reduce flash memory and data traffic required to transfer new firmware. Additionally, it also supports storing two original firmwares in the flash simultaneously, which is often used in RF testing scenarios.
- The esp_delta_ota component offers differential update functionality, which can significantly reduce the traffic required to transmit new firmware, but it does not reduce flash memory.
- The esp_bootloader_plus component only supports the differential upgrade scheme on ESP32-C2 and ESP32-C3 chips, and is no longer actively maintained.

4.9.108 Can the crystal oscillator glitch detector be disabled via software?

Yes, it can be disabled in the bootloader. However, this feature is enabled by default and needs to be manually disabled in the bootloader. The specific operation is as follows: REG_CLR_BIT(RTC_CNTL_ANA_CONF_REG, RTC_CNTL_GLITCH_RST_EN);

4.9.109 Do ESP32-C2 and ESP32-C3 built-in flash support the suspend function? After enabling suspend in the configuration, the log shows `Flash suspend feature is enabled`. What should I do?

Yes, supported. The log issue indicates that the current esp-idf version being used does not support flash suspend for XMC-D. Suggested solutions:

(1) Upgrade esp-idf to a version that supports XMC-D flash suspend (and manually enable `CONFIG_SPI_FLASH_FORCE_ENABLE_XMC_C_SUSPEND`). The versions supporting XMC-D flash suspend are as follows:

ESP-IDF Branch	Supported TAG version for XMC-D flash suspend
Release v5.4	v5.4 or later
Release v5.3	v5.3.3 or later
Release v5.2	v5.2.6 or later
Release v5.1	v5.1.7 or later
Release v5.0	v5.0.9 or later

(2) For customers who use the v4.x esp-idf branches that have already reached EOF (End-of-Life), you can fix this issue on the existing version. The main changes include:

- Add flash ID support for XMC-D to prevent exceptions when flash suspend is enabled
- Enable XMC-D flash suspend in the code

The modifications you need for esp-idf v4.4.8 are:

```
diff --git a/components/spi_flash/spi_flash_chip_generic.c
b/components/spi_flash/spi_flash_chip_generic.c
index 6726c4feeaef..4f3a35f79b11 100644
--- a/components/spi_flash/spi_flash_chip_generic.c
+++ b/components/spi_flash/spi_flash_chip_generic.c
@@ -590,6 +590,10 @@ spi_flash_caps_t
spi_flash_chip_generic_get_caps(esp_flash_t *chip)
    spi_flash_caps_t caps_flags = 0;
    // 32M-bits address support

+    if (chip->chip_id >> 16 == 0x46) {
+        caps_flags |= SPI_FLASH_CHIP_CAP_SUSPEND;
+    }
+
    // flash suspend support
    // Only `XMC` support suspend for now.
    if (chip->chip_id >> 16 == 0x20) {
@@ -786,7 +790,7 @@ esp_err_t spi_flash_common_set_io_mode(esp_flash_t *chip,
esp_flash_wrsr_func_t esp_err_t spi_flash_chip_generic_suspend_cmd_conf(esp_flash_
↪t *chip)
{
    // Only XMC support auto-suspend-
-    if (chip->chip_id >> 16 != 0x20) {
+    if (chip->chip_id >> 16 != 0x20 && chip->chip_id >> 16 != 0x46) {
        ESP_EARLY_LOGE(TAG, "The flash you use doesn't support auto suspend,
only \'XMC\' is supported");
        return ESP_ERR_NOT_SUPPORTED;
    }
--
```

4.9.110 Is UART wakeup reliable in Light Sleep mode?

The wakeup time of UART is relatively short, but the first few bits may be lost. It is not recommended for critical communication. After wakeup, it is advisable to delay about 50 ms before starting data transmission.

4.9.111 Does the ESP32 series of chips support decompression?

Yes. You can refer to the example: XZ_Decompress.

4.9.112 Does ESP-IDF v5.5.3 support ESP32-P4 v1.3?

By default, ESP-IDF v5.5.3 sets the minimum chip revision requirement for ESP32-P4 (*ESP32P4_REV_MIN*) to v3.1 (*ESP32P4_REV_MIN_301*).

Menu path: `(Top)` > `Component config` > `Hardware Settings` > `Chip revision` > `Minimum Supported ESP32-P4 Revision (Rev v3.1)`

If you are using a v1.3 chip, building and flashing directly will result in the following error:

```
A fatal error occurred: bootloader/bootloader.bin requires chip revision␣
↪in range [v3.1 - v3.99] (this chip is revision v1.3). Use --force to␣
↪flash anyway.
```

You need to enable `CONFIG_ESP32P4_SELECTS_REV_LESS_V3=y` in menuconfig, then build and flash to run properly on the v1.3 chip.

Menu path: `(Top) > Component config > Hardware Settings > Chip revision > [*] Select ESP32-P4 revisions <3.0 (No >=3.x Support)`

4.10 Thread

4.10.1 Does ESP32-H2, as an OpenThread RCP device, support building a Thread border router based on a Linux host?

- Yes. ESP32-H2 can build an OpenThread border router via a Linux host. For detailed steps, please refer to: OpenThread > Guides > Border Router > Raspberry Pi.
- On the Linux host side, please run the ot-br-posix SDK; on the ESP32-H2 device side, please run the esp-idf/examples/openthread/ot_rcp example.
- When configuring the OpenThread RCP device on ESP32-H2, please set the UART baud rate to 460800 bps. The configuration command is as follows:

```
OTBR_AGENT_OPTS="-I wpan0 -B wlan0 spinel+hdlc+uart:///dev/ttyUSB2?uart-
↪baudrate=460800"
```

- In addition, we also provide a solution for building a Thread border router based on the FreeRTOS system. For details, please refer to OpenThread > Guides > Border Router > FreeRTOS.

4.10.2 Does the ESP chip equipped with an IEEE 802.15.4 module and supporting Thread allow interaction in the form of AT-like commands?

Yes. OpenThread has released a public open-source command line interface, which can be used for networking and parameter configuration. To try a specific demo, please refer to the ot_cli example.

4.11 Wi-Fi

4.11.1 Do ESP32 and ESP8266 support Chinese SSID for Wi-Fi?

Both ESP32 and ESP8266 support Chinese SSID, but you need to use corresponding libraries and implement some settings. It should be noted you need to make special configurations when using Chinese SSID as Chinese characters occupy different numbers of bytes.

For ESP32, you can use the Wi-Fi related API provided by ESP-IDF. When connecting AP, you can use the function esp_wifi_set_config() to set Wi-Fi. The SSID parameter can be set to Chinese characters. For example,

```
wifi_config_t wifi_config = {
  .sta = {
      .ssid = "你好，世界",
      .password = "password123",
  },
};
ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config));
```

4.11.2 How much time does an ESP32 scan take?

The total time for scanning depends on:

- Active scan (by default) or passive scan.
- The time spent on each channel is 120 ms for active scanning and 360 ms for passive scanning.
- The country code and configured channel range from 1~13 channels (by default).
- Fast scan (by default) or full-channel scan.
- Station mode or Station-AP mode, and if any active connections are currently maintained.

By default, channels 1 to 11 use active scans, and channels 12 to 13 use passive scans.

- In the absence of connection in Station mode, the total time for a full-channel scan is: 11*120 + 2*360 = 2040 ms.
- With active connections in Station mode or Station-AP mode, the total time for a full-channel scan is: 11*120 + 2*360 + 13*30 = 2430 ms.

4.11.3 [Scan] Do Espressif's products support boundary scans?

No, they don't.

4.11.4 What is the definition for Wi-Fi channel? Can I select any channel of my choice?

- A Wi-Fi channel is a frequency range used for wireless communication. Different countries and regions have regulations on the available Wi-Fi channels. For example, in North America, the Wi-Fi channel ranges from 1 to 11, while in Europe, the Wi-Fi channel ranges from 1 to 13. For more details, please refer to ESP8266 Wi-Fi Channel Selection Guidelines.

4.11.5 [LWIP] With ESP-IDF v4.1, how to configure ESP32's IP address when it is in SoftAP mode?

Since ESP-IDF v4.1 and later versions do not have TCP/IP interfaces anymore, it is recommended to use the ESP-NETIF interface instead.

Code example:

```
{
    ...
    esp_netif_t *ap_netif = esp_netif_create_default_wifi_ap();
    char* ip= "192.168.5.241";
    char* gateway = "192.168.5.1";
```

(continues on next page)

(continued from previous page)

```
    char* netmask = "255.255.255.0";
    esp_netif_ip_info_t info_t;
    memset(&info_t, 0, sizeof(esp_netif_ip_info_t));

    if (ap_netif)
    {
        ESP_ERROR_CHECK(esp_netif_dhcps_stop(ap_netif));
        info_t.ip.addr = esp_ip4addr_aton((const char *)ip);
        info_t.netmask.addr = esp_ip4addr_aton((const char *)netmask);
        info_t.gw.addr = esp_ip4addr_aton((const char *)gateway);
        esp_netif_set_ip_info(ap_netif, &info_t);
        ESP_ERROR_CHECK(esp_netif_dhcps_start(ap_netif));
    }
    ...
}
```

4.11.6 [LWIP] How to configure ESP32's static IP when it is in Station mode ?

Since ESP-IDF v4.2 and later versions do not have tcp/ip interfaces anymore, it is recommended to use the ESP-NETIF interface instead. The code example is as follows:

```
esp_netif_t *sta_netif = esp_netif_create_default_wifi_sta();
if (sta_netif)
{
    esp_netif_ip_info_t info_t = {0};
    esp_netif_dhcpc_stop(sta_netif);

    info_t.ip.addr = ESP_IP4TOADDR(192, 168, 3, 23);
    info_t.gw.addr = ESP_IP4TOADDR(192, 168, 3, 1);
    info_t.netmask.addr = ESP_IP4TOADDR(255, 255, 255, 0);
    esp_netif_set_ip_info(sta_netif, &info_t);
}
esp_netif_dns_info_t dns_info = {0};
```

4.11.7 [LWIP] How to configure the Option contents of DHCP Server in ESP-IDF?

Since ESP-IDF v4.1 and later versions do not have TCP/IP interfaces anymore, it is recommended to use the ESP-NETIF interface instead. You can also refer to this example when dealing with DHCP Client configuration. The code example is as follows:

```
// Set up the handle for softap netif
esp_netif_t *ap_netif = esp_netif_create_default_wifi_ap();

// ESP_NETIF_IP_ADDRESS_LEASE_TIME, DHCP Option 51, Set the lease time␣
↪for distributed IP address
uint32_t dhcps_lease_time = 60; // The unit is min
ESP_ERROR_CHECK(esp_netif_dhcps_option(ap_netif,ESP_NETIF_OP_SET,ESP_
↪NETIF_IP_ADDRESS_LEASE_TIME,&dhcps_lease_time,sizeof(dhcps_lease_
↪time)));

// ESP_NETIF_DOMAIN_NAME_SERVER , DHCP Option 6, Set DNS SERVER
// Set the local domain DNS first
esp_netif_dns_info_t dns_info = {0};
dns_info.ip.u_addr.ip4.addr = ESP_IP4TOADDR(8,8,8,8);
```

(continues on next page)

(continued from previous page)

```
ESP_ERROR_CHECK(esp_netif_set_dns_info(ap_netif,ESP_NETIF_DNS_MAIN,&dns_
↪info));

uint8_t dns_offer = 1; // Pass 1 to make the modified DNS take effect, if␣
↪it is 0, then it means the gw ip of softap is used as the DNS server (0␣
↪by default)
ESP_ERROR_CHECK(esp_netif_dhcps_option(ap_netif,ESP_NETIF_OP_SET,ESP_
↪NETIF_DOMAIN_NAME_SERVER,&dns_offer,sizeof(dns_offer)));

// ESP_NETIF_ROUTER_SOLICITATION_ADDRESS, DHCP Option 3 Router, Pass 0 to␣
↪make the DHCP Option 3(Router) un-shown (1 by default)
uint8_t router_enable = 0;
ESP_ERROR_CHECK(esp_netif_dhcps_option(ap_netif,ESP_NETIF_OP_SET,ESP_
↪NETIF_ROUTER_SOLICITATION_ADDRESS,&router_enable, sizeof(router_
↪enable)));

// ESP_NETIF_SUBNET_MASK, DHCP Option 1, Configure the subnet mask
// If it fails to configure the subnet mask via ESP_NETIF_SUBNET_MASK,␣
↪please make modifications using esp_netif_set_ip_info
```

4.11.8 [Performance] How to test the bit rate of Wi-Fi modules?

Please use the example iperf in ESP-IDF for testing.

4.11.9 [LWIP] What is the default IP address of ESP8266 SoftAP?

The default network segment used by ESP8266 SoftAP is 192.168.4.*.

4.11.10 Why do I have problem connecting to router with IP 192.168.4.X in SoftAP + Station mode?

The default network segment used by ESP8266 SoftAP is 192.168.4.*, and its IP address is 192.168.4.1. When connecting ESP8266 to the router of 192.168.4.X, it cannot distinguish whether this address indicates its own SoftAP or the external router.

4.11.11 [Connect] How many devices is ESP8266 able to connect in SoftAP mode?

The ESP8266 chip in SoftAP mode supports connecting eight devices at most. This is because the NAT (Network Address Translation) mechanism used by the ESP8266 chip in SoftAP mode only supports eight devices at most. However, it should be noted that each connected device will occupy a certain amount of bandwidth and resources. Therefore, we recommend connecting four devices as too many devices may affect the performance and stability of the Wi-Fi module.

4.11.12 Do ESP8266/ESP32/ESP32-S2/S3/C2/C3 support web/SoftAP provisioning?

Yes.

- For ESP8266, please refer to example ESP8266 softap_prov.
- For ESP32/ESP32-S2/S3/C2/C3, please refer to example ESP32/ESP32-S2/S3/C2/C3 wifi_prov_mgr.

4.11.13 [Connect] How do ESP8266 and ESP32 hide SSID in SoftAP mode?

To hide ESP8266 or ESP32 as SSID in SoftAP mode, you can use the following methods:

Use the esp_wifi_set_config() function to configure the SSID and password in SoftAP mode and whether to hiding them. For example, the following code sets the SSID to "MySoftAP", the password to "MyPassword", and set .ssid_hidden = 1 to hide the SSID:

```
wifi_config_t config = {
  .ap = {
    .ssid = "MySoftAP",
    .ssid_len = strlen("MySoftAP"),
    .password = "MyPassword",
    .max_connection = 4,
    .authmode = WIFI_AUTH_WPA_WPA2_PSK
    .ssid_hidden = 1
  },
};
esp_wifi_set_config(WIFI_IF_AP, &config);
```

Then use esp_wifi_start() function to starts Wi-Fi.

4.11.14 Does the buffer parameter in esp_wifi_802.11_tx interface include FCS?

No, the FCS frame is generated automatically by hardware.

4.11.15 What is the supported Wi-Fi frequency band and power meter for ESP32?

The Wi-Fi frequency band is 2412 ~ 2484 MHz, and the available channels and corresponding operating frequencies can be configured in software. There are default values in power meter, and it can also be configured by software. For detailed guidance, please refer to Power Limit Tool.

4.11.16 What is the maximum value of ESP32 Wi-Fi RF power ?

The output RF power of ESP32 can be set to 20 dBm at maximum. Please note that the maximum output power may vary in different countries and regions. Please ensure that you comply with local rules and regulations when using ESP32. In addition, high power output also influence battery life and Wi-Fi signal stability. As a result, you should confirm the output power depending on applications and requirements.

4.11.17 How does ESP32 adjust Wi-Fi TX power?

- Configure `Component config > PHY > Max Wi-Fi TX power(dBm)` via menuconfig, and the max value is 20 dBm.
- Use API *esp_err_t esp_wifi_set_max_tx_power(int8_t power);*.

4.11.18 [Connect] How many devices is ESP32 able to connect in AP mode?

Up to 10 devices in AP mode. It is configured to support four devices by default.

4.11.19 [Connect] How do Wi-Fi modules rank signal strength levels based on RSSI values ?

We do not have a rating for RSSI signal strength. You can take the calculation method from Android system for reference if you need a standard for classification.

```
@UnsupportedAppUsage
private static final int MIN_RSSI = -100;

/** Anything better than or equal to this will show the max bars. */
@UnsupportedAppUsage
private static final int MAX_RSSI = -55;

public static int calculateSignalLevel(int rssi, int numLevels) {
  if(rssi <= MIN_RSSI) {
    return 0;
  } else if (rssi >= MAX_RSSI) {
    return numLevels - 1;
  } else {
    float inputRange = (MAX_RSSI -MIN_RSSI);
    float outputRange = (numLevels - 1);
    return (int)((float)(rssi - MIN_RSSI) * outputRange / inputRange);
  }
}
```

4.11.20 [Connect] Why does ESP32 disconnect from STA when it is in Soft-AP mode?

- By default, the ESP32 will disconnect from the connected STA if it doesn't receive any data from this STA for continuous 5 minutes. This time can be modified via API esp_wifi_set_inactive_time.
- Note: esp_wifi_set_inactive_time is a newly added API.
  - master commit: `63b566eb27da187c13f9b6ef707ab3315da24c9d`
  - 4.2 commit: `d0dae5426380f771b0e192d8ccb051ce5308485e`
  - 4.1 commit: `445635fe45b7205497ad81289c5a808156a43539`
  - 4.0 commit: `0a8abf6ffececa37538f7293063dc0b50c72082a`
  - 3.3 commit: `908938bc3cd917edec2ed37a709a153182d511da`

4.11.21 [Connect] While ESP32 connecting Wi-Fi, how can I determine the reason of failure by error codes?

For ESP-IDF v4.0 and later versions, please refer to the following codes to get the reason:

```
if (event_base == WIFI_EVENT && event_id == WIFI_EVENT_STA_DISCONNECTED) {
  wifi_event_sta_disconnected_t *sta_disconnect_evt = (wifi_event_sta_
↪disconnected_t*)event_data;
  ESP_LOGI(TAG, "wifi disconnect reason:%d", sta_disconnect_evt->reason);
  esp_wifi_connect();
  xEventGroupClearBits(s_wifi_event_group, CONNECTED_BIT);
}
```

When the callback function receives the `WIFI_EVENT_STA_DISCONNECTED` event, the failure reason can be obtained through the `reason` variable of the structure wifi_event_sta_disconnected_t.

- `WIFI_REASON_AUTH_EXPIRE`: This code is returned during the auth phase when the STA sends an auth but do not received any auth reply from the AP within the specified time. The possibility of this code occurrence is low.
- `WIFI_REASON_AUTH_LEAVE`: This code is sent by AP, normally because the AP is disconnected from the STA for some reason.
- `WIFI_REASON_4WAY_HANDSHAKE_TIMEOUT` or `WIFI_REASON_HANDSHAKE_TIMEOUT`: Wrong password.
  `WIFI_REASON_4WAY_HANDSHAKE_TIMEOUT` is the standard generalized error code, while `WIFI_REASON_HANDSHAKE_TIMEOUT` is a customized error code. The main difference is: `WIFI_REASON_4WAY_HANDSHAKE_TIMEOUT` occurs when the router tells the device the password is wrong; `WIFI_REASON_HANDSHAKE_TIMEOUT` occurs when the device itself performs a timeout mechanism without being informed about the wrong password by the router.
- `WIFI_REASON_CONNECTION_FAIL`: This code is returned during the scan phase when the STA scanned a matched AP while the AP is in the blacklist. This is probably because that the AP has actively disconnected from the STA last time or something wrong happened when the STA connecting the AP.

4.11.22 Does ESP32 perform domain name resolution each time it connects to the server?

The domain name is resolved via DNS within the stack, and the resolved data will be cached within the specified time. The cache time is based on the TTL data obtained from the DNS server, which is a parameter filled when configuring the domain name, usually 10 minutes.

4.11.23 [Connect] What does the number after the state machine switch in Wi-Fi log mean?

eg: run -> init (fc0), fc0 means the STA has received the deauth frame and reason is password error.

- c0 indicates the received frame type (00 indicates a timeout)
- f indicates reason

Frame type: [a0 disassoc], [b0 auth], [c0 deauth].

4.11.24 [Connect] What does bcn_timeout, ap_probe_send_start mean ?

The STA does not receive the Beacon frame within the specified time (6 s by default for ESP32, equals to 60 Beacon Intervals). - The reason could be:

- Insufficient memory. "ESP32_WIFI_MGMT_SBUF_NUM" is not enough (there will be errors like "esf_buf: t=8, l=beacon_len, ···" in the log). You can check this by typing the heap size when received a Disconnect event.
- The AP did not send a beacon. This can be checked by capturing beacons from AP.
- Rssi too low. When the Rssi value is too low in complex environments, the STA may not receive the beacon. This can be checked by retrieving Rssi values via `esp_wifi_sta_get_ap_info`.
- Hardware related issues. Bad package capturing performance.

When there is a bcn_timeout, the STA will try to send Probe Request for five times. If a Probe Response is received from the AP, the connection will be kept, otherwise, the STA will send a Disconnect event and the connection will fail.

4.11.25 [Connect] How to reconnect Wi-Fi after it disconnected?

Call esp_wifi_connect after received the `WIFI_EVENT_STA_DISCONNECTED` event.

4.11.26 [Connect] When does ESP32 disconnect from SoftAP in station mode ?

By default, the ESP32 will disconnect from the AP if it does not receive any beacon for 6 s. This time can be modified via esp_wifi_set_inactive_time.

4.11.27 [Scan] Why does the STA cannot find any AP sometimes during the scanning?

There are many possible reasons why ESP32 and ESP8266 cannot scan any AP, and some common reasons and solutions are listed below.

- The AP is too far away or the signal is too weak, while Wi-Fi of ESP32 and ESP8266 can only work within a certain range. If the AP is too far away or the Wi-Fi signal is too weak, ESP32 and ESP8266 may not be able to scan the AP. You can move ESP32 or ESP8266 closer to the AP or use a signal amplifier to enhance the signal strength.
- SSID of the AP is hidden. SSID of some APs may be hidden, so they will not be broadcasted to nearby devices. In this case, ESP32 and ESP8266 cannot scan these APs. You can connect these APs by inputting their SSID and passwords manually.
- The AP is overloaded or malfunctioning. If the AP is overloaded or malfunctioning, it may not be able to handle new connection requests, and thus ESP32 and ESP8266 cannot connect to the AP. You can try to wait for a while and then scan the AP again.
- ESP32 or ESP8266 has some software issues. Sometimes, software issues of ESP32 or ESP8266 may cause problems with scanning for APs. In this case, you can try to reset ESP32 or ESP8266 and restart Wi-Fi. If this method does not work, you may need to update the firmware of ESP32 or ESP8266.
- Other reasons include wireless interference, security settings, and network configuration. These reasons may also affect Wi-Fi of ESP32 or ESP8266. In this case, you need to carefully check the Wi-Fi environment and implement corresponding settings.

4.11.28 [Scan] What is the maximum number of APs that can be scanned ?

There is no limit to the maximum number of APs that can be scanned. The number depends on how many APs are around and configurations of the scanning parameters, such as the time spent on each channel, the longer time spent on each channel the more likely it is to find all the APs.

4.11.29 [Scan] Can I choose to connect the best AP when there are multiple APs with identical ssid/password during the scan ?

By default, the scan type is WIFI_FAST_SCAN, which makes the STA always connects the first AP during the scan. If you expect to connect the best AP, please set scan_method to WIFI_ALL_CHANNEL_SCAN and configure sort_method to determine whether to choose the AP with the strongest RSSI or connect to the most secure AP.

4.11.30 [Scan] How to configure scan_method in the wifi_sta_config_t structure? What is the difference between all_channel_scan and fast_scan?

all_channel_scan and fast_scan are used to find the appropriate AP before connecting. The scan_method is set to fast_scan by default, which is mainly used together with threshold to filter APs with weak signal or encryption.

- When fast_scan is set, the STA will stop scanning once it finds the first matched AP and then connect to it, so as to save time for connection.
- When all_channel_scan is set, the STA will scan all channels and store four APs with the best signal or the most secure encryption according to the sorting method configured in sort_method. After the scan is completed, the STA will connect one of the four APs with the best signal or the most secure encryption.

4.11.31 [LWIP] How to get error code of the socket?

- For ESP-IDF v4.0 and later versions: use the value of *errno* directly to get the error code after the socket API returns failure.
- For previous versions of ESP-IDF v4.0: call *getsockopt(sockfd, SOL_SOCKET, SO_ERROR, ⋯)* to get the error code after the socket API returns failure, otherwise you may get wrong error code when multiple sockets operate simultaneously.

4.11.32 [LWIP] What is the default keep-alive time of TCP?

By default, a TCP keep-alive message will be sent every 75 seconds for 9 times if no TCP message is received for two consecutive hours. Then, if there is still no message received, the LWIP will disconnect from the TCP.

The keep-alive time can be configured via socket option.

4.11.33 [LWIP] What is the retransmission interval of TCP ?

When ESP32 acts as the sender, the initial retransmission interval of the TCP protocol is usually set to 3 seconds. If the receiver does not send an ACK message, the next retransmission interval will be determined based on the Jacobson algorithm. The retransmission interval will be exponentially increased. Specially, it will be increased by 2, 4, 8, 16, 32 seconds gradually. This retransmission interval is not fixed and you can adjust it by changing some parameters such as timeout time and the size of sliding window.

4.11.34 [LWIP] What is the maximum number of sockets that can be created?

Up to 16, default is 10.

4.11.35 [Sleep] What kinds of sleeping mode does ESP32 have? What are the differences?

- There are mainly three sleeping modes: Modem sleep, Light sleep and Deep sleep.
  - Modem sleep: the station WMM sleeping mode specified in the Wi-Fi protocol (the station sends NULL data frame to tell the AP to sleep or wake up). The Modem sleep mode is enabled automatically after the station connected to AP. After entering this mode, the RF block is disabled and the station stays connected with the AP. The Modem sleep mode will be disabled after the station disconnected from the AP. The ESP32 can also be configured to decrease the CPU' s clock frequency after entering Modem sleep mode to further reduce its current.
  - Light sleep: this is a station sleep mode based on Modem sleep mode. The difference between is that, besides for the RF block being disabled, the CPU will also be suspended in this mode. After exiting from Light sleep mode, the CPU continues to operate from where it stopped.
  - Deep sleep: a sleeping mode un-specified in the Wi-Fi protocol. During Deep sleep mode, all the blocks except for RTC is disabled, and the station cannot be connected to AP. After exiting from this mode, the whole system will restart to operate (similar to system restart).

4.11.36 [Sleep] How can I enable the dynamic frequency scaling function for ESP32 in Modem-sleep mode?

You can enable this function by selecting `menuconfig > Component Config > Power Management > Support for power management > Enable dynamic frequency scaling (DFS) at startup (NEW)`.

4.11.37 [Sleep] How low can the speedstep function go for ESP32 in modem sleep mode ?

For now, the CPU clock can go down to as low as 40 MHz.

4.11.38 [Sleep] What affects the average current of ESP32 in modem sleep mode?

The modem sleep mode of ESP32 is achieved by setting wakeup cycles. ESP32 opens RF functions to communicate during every cycle and closes these functions in the rest of the time.

The average current of this mode is influenced by several factors, including:

- Wakeup cycle: A shorter wakeup cycle means the chip will be waked up more frequently, which increases the average current.
- Signal quality: If the Wi-Fi signal is weak, the chip will keep trying to reconnect or send data, which will increase the average current. Using communication protocols with bigger transmission power will also increase the average current.
- Hardware configuration: The hardware configuration of the chip also affects power consumption, such as the number of CPU cores, CPU clock frequency, CPU idle time ratio, power supply voltage, external crystal oscillator, etc. All of these factors influence the size of average current.
- Other factors include whether the testing router accurately sends beacon timestamps, whether too many broadcast packets have been sent, whether peripheral modules are working, etc.

4.11.39 [Sleep] Why the average current measured in modem sleep mode is a bit high?

- A lot of Wi-Fi data sent and received during the test. The more data there is, the less chance there will be for entering sleeping mode and the higher average current will be.
- The time when the router sends out beacon is not accurate. The station needs to wake up and monitor the beacon regularly, thus it will wait longer if the beacon time is not accurate. In this way, the station has less time in sleeping mode and the average current will be high.
- There are peripheral modules working during the test. Please close them before the test.
- The station+SoftAP mode is enabled. During modem sleep state, the current will only be lower in station-only mode.

4.11.40 [Sleep] Why the average current measured in light sleep mode is a bit high?

Besides for the reasons listed in the last question, the possible reasons also could be:

- The application layer code is running continuously, thus the CPU does not get chance to suspend.
- The application layer has enabled ets timer or esp timer and the timeout interval is short, thus the CPU does not get chance to suspend.

4.11.41 [Sleep] What kinds of power-saving modes does ESP32 have? What are the differences?

There are mainly three modes: minimum modem power-saving, maximum modem power-saving, and no power save modes.

- Minimum modem: default type. In this mode, the station wakes up to receive beacon every DTIM period, which is equal to (DTIM * 102.4) ms. For example, if the DTIM of the router is 1, the station will wake up every 100 ms.
- Maximum modem: in this mode, the interval to receive beacons is determined by the `listen_interval` parameter in `wifi_sta_config_t`. The interval is equal to (listen interval * 102.4) ms. For example, if the DTIM of the router is 1, and the listen interval is 10, the station will wake up every 1 s.
- No power save: no power save.

4.11.42 Does ESP8266 support 802.11k/v/r protocol?

For now, the ESP8266 only supports 802.11k and 802.11v, please refer to example roaming.

4.11.43 Does ESP32 Wi-Fi support roaming between different APs with the same SSID?

Yes, currently it supports 802.11k and 802.11v protocols. Please refer to the example roaming.

4.11.44 [Connect] After the NONOS_SDK updated from version *2.1.0* to *2.2.2*, why does the connecting time become longer ?

Please update to version *master*, which has solved the incompatibility issue between the CCMP encryption and some APs.

4.11.45 How does ESP32 receive and transmit Wi-Fi 802.11 packets?

- By using the following APIs:

```
esp_err_t esp_wifi_80211_tx(wifi_interface_t ifx, const void *buffer, int␣
↪len, bool en_sys_seq);
esp_wifi_set_promiscuous_rx_cb(wifi_sniffer_cb);
```

- The abovementioned APIs are also used in the ESP-MDF project, please refer to mconfig_chain.

4.11.46 [Connect] The ESP32 and ESP8266 failed to connect to router, what could be the reasons ?

- Please check if the SSID or password is wrong.
- There could be errors in different Chinese codes, so it is not recommended to use an SSID written in Chinese.
- The settings of bssid_set. If the MAC address of the router does not need to be identified, the stationConf.bssid_set should be configured to 0.
- It is recommended to define the wifi_config field in wifi_config_t using the static variable *static*.

4.11.47 [Connect] What kind of networking methods does ESP8266 have ?

- SmartConfig mode: using SmartConfig. The device scans feature pack in sniffer mode.
- SoftAP mode: the device enables SoftAP and sends SSID and password after the phone connects to SoftAP and set up a stable TCP/UDP connection.
- WPS mode: an additional button should be added on the device; or using the phone to enable WPS after it connected to SoftAP.

4.11.48 [Connect] What are the specifications of Wi-Fi parameters when using SmartConfig?

SmartConfig is a method of configuring Wi-Fi parameters via local network broadcasting. Users can send Wi-Fi account and password to the device through a matching APP. SmartConfig network configuration has some requirements on Wi-Fi parameters:

- SSID name: Supports Chinese characters and English letters, digits, with a maximum length of 32 bytes.
- Wi-Fi password: 8-64 digits, case-sensitive.
- Wi-Fi security encryption: Currently, SmartConfig supports WPA, WPA2, and WEP encryption methods, and does not support open methods without encryption.

4.11.49 [Connect] Does ESP8266 Wi-Fi support WPA2 enterprise-level encryption ?

- Yes. Please refer to example wpa2_enterprise.
- To build RADIUS server, you can use FreeRADIUS.

4.11.50 [Connect] What are the low-power modes for ESP32 to maintain its connection to Wi-Fi?

- In such scenarios, the chip switches between Active mode and Modem-sleep mode automatically, making the power consumption also varies in these two modes.
- The ESP32 supports Wi-Fi keep-alive in Light-sleep mode, and the auto wakeup interval is determined by the DTIM parameter.
- Please find examples in ESP-IDF - > examples - > wifi - > power_save.

4.11.51 Do Espressif' s chips support WPA3?

- ESP32 series: WPA3 is supported from esp-idf release/v4.1 and enabled by default. Go to menuconfig > Component config > Wi-Fi for configuration.
- ESP8266: WPA3 is supported from the release/v3.4 branch of ESP8266_RTOS_SDK and enabled by default. Go to menuconfig > Component config > Wi-Fi for configuration.

4.11.52 [Connect] How does the device choose AP when there are multiple identical SSIDs in the current environment?

- The device connects to the first scanned AP.
- If you expect to sort APs by signal quality and etc., use the scan function to filter manually.
- If you expect to connect to a specified AP, add BSSID information in connection parameters.

4.11.53 [Connect] Does ESP8266 have repeater solutions?

- We have not officially released such application solutions yet.
- For relay related applications, please find on github. The relay rates should be set basing on real tests.

4.11.54 What is the retransmission time for ESP32' s data frame and management frame ? Can this be configured ?

The retransmission time is 31 and it can not be configured.

4.11.55 How to customize the hostname for ESP32 ?

- Taking ESP-IDF V4.2 as an example, you can go to menuconfig > Component Config > LWIP > Local netif hostname, and type in the customized hostname.
- There may be a slight difference on naming in different versions.

4.11.56 How to obtain 802.11 Wi-Fi packets ?

- Please refer to Wireshark User Guide in ESP-IDF Programming Guide.
- Please note that the wireless network interface controller (WNIC) that you use should support the Monitor mode.

4.11.57 Does ESP32 Wi-Fi support PMF (Protected Management Frames) and PFS (Perfect Forward Secrecy) ?

The PMF is supported both in WPA2 and WPA3, and PFS is supported in WPA3.

4.11.58 Why does ESP8266 print out an AES PN error log when using esptouch v2?

This occurs when ESP8266 has received retransmitted packets from the router for multiple times. However, this will not affect your usage.

4.11.59 Does ESP32 WFA certification support multicast?

No. It is recommended to refer to the ASD-1148 method of testing.

4.11.60 When using ESP32 to establish a hotspot, can I scan all APs and the occupied channels first, and then select the smallest and cleanest channel to establish my own AP?

- You can scan all APs and occupied channels before establishing a hotspot. Refer to the API esp_wifi_scan_get_ap_records.
- It cannot be performed automatically. You need to customize the channel selection algorithm to implement such operation.

4.11.61 I' m scanning Wi-Fi on an ESP32 device using release/v3.3 version of ESP-IDF. When there are some identical SSIDs, same SSID names will show in the Wi-Fi list repeatedly. Is there an API to filter such repeated names?

No, same SSID names cannot be filtered out since identical SSID names may not mean identical servers. Their BSSID may not be the same.

4.11.62 Does ESP8266 support EDCF (AC) scheme?

The master version of ESP8266-RTOS-SDK supports EDCF (AC) applications, but no application examples are provided for now. You can enable Wi-Fi QoS configuration in `menuconfig -> Component config -> Wi-Fi` to get support.

4.11.63 I' m using the master version of ESP8266-RTOS-SDK to open the WiFi Qos application to get EDCF support. How does ESP8266 decide which data packet should be allocated to the EDCF AC category?

It can be determined by setting `IPH_TOS_SET(iphdr, tos)`.

4.11.64 Using ESP32, how to configure the maximum Wi-Fi transmission speed and stability without considering memory and power consumption?

To configure the maximum Wi-Fi transmission speed and stability, please refer to How to improve Wi-Fi performance in ESP-IDF programming guide and set the relevant configuration parameters in `menuconfig`. The option path can be found by searching "/" in the `menuconfig` interface. The optimal configuration parameters need to be tested according to the actual environment.

4.11.65 In Wi-Fi SoftAP mode, how many Station devices can ESP8266 be connected at most?

ESP8266 supports up to 8 Station device connections.

4.11.66 How to get CSI data when using ESP32 device in Station mode?

- CSI data can be obtained by calling 'esp_wifi_set_csi_rx_cb()'. See description in API.
- For specific steps, please refer to Espressif CSI examples.

4.11.67 In AP + STA mode, after an ESP32 is connected to Wi-Fi, will the Wi-Fi connection be affected if I enable or disable its AP mode?

After an ESP32 is connected to Wi-Fi in AP + STA dual mode, AP mode can be enabled or disabled at will without affecting Wi-Fi connection.

4.11.68 I' m using ESP-IDF release/v3.3 for ESP32 development, but only bluetooth function is needed, how to disable Wi-Fi function through software?

- Please call `esp_wifi_stop()` to disable the Wi-Fi function. For API description, please see esp_err_t esp_wifi_stop(void).
- If you need to reclaim the resources occupied by Wi-Fi, call `esp_wifi_deinit()`. For API description, please see esp_err_t esp_wifi_deinit(void).
- Here is a simple example:

```
#include "esp_wifi.h"
#include "esp_bt.h"

void app_main()
{
  // Turn off Wi-Fi functionality
  esp_wifi_stop();

  // Initializing Bluetooth functionality
  esp_bt_controller_config_t bt_cfg = BT_CONTROLLER_INIT_CONFIG_DEFAULT();
  esp_bt_controller_init(&bt_cfg);
  esp_bt_controller_enable(ESP_BT_MODE_BTDM);

  // ...
}
```

In this example, the esp_wifi_stop() function is called to turn off Wi-Fi and then to initialize Bluetooth. It should be noted that once Wi-Fi is stopped, Wi-Fi related APIs cannot be used.

4.11.69 In ESP-IDF, the `esp_wifi_80211_tx()` interface can only be used to send data packets, is there a corresponding function to receive packets?

- Please use callback function to received data packets as follows:

```
esp_wifi_set_promiscuous_rx_cb(wifi_sniffer_cb);
esp_wifi_set_promiscuous(true);
```

- The above data receive method is also used in another open-sourced project, please see esp-mdf.

4.11.70 What are the reasons for the high failure rate of esptouch networking?

CHIP: ESP32, ESP32S2, ESP32S3, ESP32C3, ESP8266

- The same hotspot is connected too many people.
- The signal quality of the hotspot connected by cell phone is poor.
- The router does not forward multicast data.
- The router has enabled dual-band integration, and the phone is connected to the 5G frequency band.

4.11.71 How to optimize the IRAM when ESP32 uses Wi-Fi?

You can disable `WIFI_IRAM_OPT`, `WIFI_RX_IRAM_OPT` and `LWIP_IRAM_OPTIMIZATION` in menuconfig to optimize IRAM space, but this will degrade Wi-Fi performance.

4.11.72 How to test ESP32's Wi-Fi transmission distance?

You can use the iperf example and configure the ESP32 device to iperf UDP mode. Then, you can distance the device continuously to see at which point the Wi-Fi data transmission rate will drop to 0.

4.11.73 What is the maximum length of Wi-Fi MTU for an ESP32 and how to change it?

The maximum Wi-Fi MTU length for ESP32 is 1500. You can change this value in the LwIP component by `netif > mtu`. However, it is not recommended to change this value.

4.11.74 During the on-hook test for an ESP32 device, the following log shows. What does it mean?

log:

```
[21-01-27_14:53:56]I (81447377) wifi:new:<7,0>, old:<7,2>, ap:<255,255>,␣
↪sta:<7,0>, prof:1
[21-01-27_14:53:57]I (81448397) wifi:new:<7,2>, old:<7,0>, ap:<255,255>,␣
↪sta:<7,2>, prof:1
[21-01-27_14:53:58]I (81449417) wifi:new:<7,0>, old:<7,2>, ap:<255,255>,␣
↪sta:<7,0>, prof:1
[21-01-27_14:53:59]I (81450337) wifi:new:<7,2>, old:<7,0>, ap:<255,255>,␣
↪sta:<7,2>, prof:1
```

- The value after `new` represents the current primary and secondary channel; the value after `old` represents the last primary and secondary channel; and the value after `ap` represents the primary and secondary channel of the current ESP32 AP, which will be 255 if softAP is not enabled; the value after `sta` represents primary and secondary channel of the current ESP32 sta; and `prof` is the channel of ESP32's softAP stored in NVS.
- For the numerical values represented by the secondary channel, please refer to wifi_second_chan_t.
- The above log indicates that router is switching between HT20 and HT40 minus. You can check the Wi-Fi bandwidth setting of the router.

4.11.75 How to disable AP mode when ESP32 is in AP + STA mode?

- This can be done through the configuration of `esp_wifi_set_mode(wifi_mode_t mode);` function.
- Just call `esp_wifi_set_mode(WIFI_MODE_STA);`.

4.11.76 After ESP32 used the Wi-Fi function, are all ADC2 channels unavailable?

When an ESP32 device is using Wi-Fi function, the ADC2 pins that are not occupied by Wi-Fi can be used as normal GPIOs. You can refer to the official ADC Description.

4.11.77 How do I set the country code for a Wi-Fi module ?

CHIP: ESP8266 | ESP32 | ESP32 | ESP32-C3

Please call esp_wifi_set_country to set the country code.

4.11.78 When using ESP32 as a SoftAP and have it connected to an iPhone, a warning prompts as "low security WPA/WPA2(TKIP) is not secure. If this is your wireless LAN, please configure the router to use WPA2(AES) or WPA3 security type" , how to solve it?

IDF: release/v4.0 and above

- You can refer to the following code snippet:

```
wifi_config_t wifi_config = {
    .ap = {
        .ssid = EXAMPLE_ESP_WIFI_SSID,
        .ssid_len = strlen(EXAMPLE_ESP_WIFI_SSID),
        .channel = EXAMPLE_ESP_WIFI_CHANNEL,
        .password = EXAMPLE_ESP_WIFI_PASS,
        .max_connection = EXAMPLE_MAX_STA_CONN,
        .authmode = WIFI_AUTH_WPA2_PSK,
        .pairwise_cipher = WIFI_CIPHER_TYPE_CCMP
    },
};
```

- WIFI_AUTH_WPA2_PSK is AES, also called CCMP. WIFI_AUTH_WPA_PSK is TKIP. WIFI_AUTH_WPA_WPA2_PSK is TKIP+CCMP.

4.11.79 Since ESP32' s Wi-Fi module only supports 2.4 GHz of bandwidth, can Wi-Fi networking succeed when using a multi-frequency router with both 2.4 GHz and 5 GHz of bandwidth ?

Please set your router to multi-frequency mode (can support 2.4 GHz and 5 GHz for one Wi-Fi account), and the ESP32 device can connect to Wi-Fi normally.

4.11.80 How to obtain the RSSI of the station connected when ESP32 is used in AP mode?

- You can call API esp_wifi_ap_get_sta_list, please refer to the following code snippet:

```
{
    wifi_sta_list_t wifi_sta_list;
    esp_wifi_ap_get_sta_list(&wifi_sta_list);
    for (int i = 0; i < wifi_sta_list.num; i++) {
        printf("mac address: %02x:%02x:%02x:%02x:%02x:%02x\t rssi:%d\n",wifi_
↪sta_list.sta[i].mac[0], wifi_sta_list.sta[i].mac[1],wifi_sta_list.sta[i].
↪mac[2],
                  wifi_sta_list.sta[i].mac[3],wifi_sta_list.sta[i].mac[4],wifi_
↪sta_list.sta[i].mac[5],wifi_sta_list.sta[i].rssi);
    }
}
```

- The RSSI obtained by `esp_wifi_ap_get_sta_list` is the average value over a period of time, not real-time RSSI. The previous RSSI has a weight of 13, and the new RSSI has a weight of 3. The RSSI is updated when it is or larger than 100ms, the old rssi_arg is used when updating as: `rssi_avg = rssi_avg*13/16 + new_rssi * 3/16`.

4.11.81 Does ESP32 support FTM(Fine Timing Measurement)?

- No, it doesn't. FTM needs hardware support, but ESP32 doesn't have it.
- ESP32-S2 and ESP32-C3 can support FTM in hardware.
- ESP-IDF can support FTM from v4.3-beta1.
- For more information and examples of FTM, please refer to FTM.

4.11.82 When ESP32 is in STA+AP mode, how to specify whether using STA or AP interface to send data?

Background:

The default network segment of ESP32 as AP is 192.168.4.x, and the network segment of the router to which ESP32 as STA is connected is also 192.168.4.x. The PC connects to the ESP32 AP and creates a tcp server. In this case, the tcp connection between ESP32 as tcp client and PC as tcp server cannot be established successfully.

Solutions:

- It is possible for ESP32 to specify whether to use STA or AP interface for data transmission. Please see example tcp_client_multi_net, in which both ethernet and station interface are used and each can be specified for data transmission.
- There are two ways to bind socket to an interface:
  - use netif name (use socket option SO_BINDTODEVICE)
  - use netif local IP address (get IP address of an interface via esp_netif_get_ip_info(), then call bind())

Note:

- The tcp connection between ESP32 and PC can be established when an ESP32 is bound to the STA interface, while the connection cannot be established when it is bound to the AP interface.
- By default, the tcp connection between ESP32 and mobile phone can be established(the mobile phone as a station is connected to ESP32).

4.11.83 ESP8266 wpa2_enterprise How to enable Wi-Fi debugging function?

Open menuconfig via `idf.py menuconfig` and configure the following parameters:

```
menuconfig==>Component config ==>Wi-Fi ==>
[*]Enable WiFi debug log ==>The DEBUG level is enabled (Verbose)
[*]WiFi debug log submodule
[*] scan
[*] NET80211
[*] wpa
[*] wpa2_enterprise

menuconfig==>Component config ==>Supplicant ==>
[*] Print debug messages from WPA Supplicant
```

4.11.84 Is there a standard for the number of Wi-Fi signal frames?

CHIP: ESP8266 | ESP32 | ESP32 | ESP32-C3

There is no such standard for now. You can do the calculation by yourself based on the received RSSI. For example, if the received RSSI range is [0,-96], and the required signal strength is 5, then [0~-20] is the full signal, and so on.

4.11.85 What is the current progress of WFA bugs fixing?

CHIP: ESP32 | ESP32-S2 | ESP32-C3 | ESP8266

For more details, please refer to the Wi-Fi Security Notice on the Espressif official website.

4.11.86 When Wi-Fi connection failed, what does the error code mean?

CHIP: ESP32

- Any error occurred during the Wi-Fi connection will cause it coming to init status, and there will be a hexadecimal number in the log, e.g., `wifi:state, auth-> init(200)`. The first two digits indicate error reasons while the last two digits indicate the type code of the received or transmitted management frame. Common frame type codes are 00 (received nothing, timeout), A0 (disassoc), B0 (auth) and C0 (deauth).
- Error reasons indicated by the first two digits can be found in Wi-Fi Reason Code. The last two digits can be checked in frame management code directly.

4.11.87 When using ESP32's Release/v3.3 of SDK to download the Station example, the device cannot be connected to an unencrypted Wi-Fi. What is the reason?

- In the example, it is by default to connect to an encrypted AP as:

```
.threshold.authmode = WIFI_AUTH_WPA2_PSK,
```

- If you need connect to an unencrypted AP, please set the following parameter to 0:

```
.threshold.authmode = 0,
```

- For AP mode selection instructions, please refer to esp_wifi_types.

4.11.88 What is the maximum PHY rate of Wi-Fi communication of ESP32-S2 chip?

The theoretical maximum PHY rate of ESP32-S2 Wi-Fi communication is 150 Mbps.

4.11.89 Does ESP modules support EAP-FAST?

:CHIP: ESP32 | ESP32-S2 | ESP32-C3 :

Yes, please refer to wifi_eap_fast demo.

4.11.90 Does ESP modules support the WiFi NAN (Neighbor Awareness Networking) protocol?

CHIP: ESP8266 | ESP32 | ESP32-C3 | ESP32-S2 | ESP32-S3

No.

4.11.91 When using ESP32 with release/v3.3 version of ESP-IDF. When configuring the router, is there an API to directly tell that the entered password is wrong?

- There is no such API. According to the Wi-Fi protocol standard, when the password is wrong, the router will not clearly tell the Station that the 4-way handshake is due to the password error. Under normal circumstances, the password is obtained in 4 packets (1/4 frame, 2/4 frame, 3/4 frame, 4/4 frame). When the password is correct, the AP will send 3/4 frames, but when the password is wrong, the AP will not send 3/4 frame but send 1/4 frame instead. However, when the AP sends 3/4 frame which is lost in the air for some reason, the AP will also re-send 1/4 frame. Therefore, for Station, it is impossible to accurately distinguish between these two situations. In the end, it will report a 204 error or a 14 error.
- Please refer to Wi-Fi reason code.

4.11.92 When testing the Station example of ESP32 base on v4.4 version of ESP-IDF, how to support WPA3 encryption mode?

- Open `menuconfig → Component config → Wi-Fi → Enable WPA3-Personal` configuration;
- Set `capable = true` in `pmf_cfg` in the application code;
- Please refer to Wi-Fi Security for more descriptions.

4.11.93 How does ESP32 speed up the Wi-Fi connection?

You can try the following approaches:

- Set the CPU frequency to the maximum to speed up the key calculation speed. In addition, you can also set the flash parameters to `QIO, 80MHz`, which will increase power consumption.
- Disable `CONFIG_LWIP_DHCP_DOES_ARP_CHECK` to greatly reduce the time of getting IP. But there will be no checking on whether there is an IP address conflict in the LAN.
- Open `CONFIG_LWIP_DHCP_RESTORE_LAST_IP`, and save the IP address obtained last time. When DHCP starts, send DHCP requests directly without performing DHCP discover.
- Use fixed scanning channel.

4.11.94 Does ESP32 WPA2 Enterprise Authentication support Cisco CCKM mode?

This mode is currently not supported, even though the enumeration in esp_wifi_driver.h has WPA2_AUTH_CCKM.

4.11.95 Using wpa2_enterprise (EAP-TLS method), what is the maximum length supported for client certificates?

Up to 4 KB.

4.11.96 Does ESP8089 support Wi-Fi Direct mode?

Yes, but ESP8089 can only use the default fixed firmware and cannot be used for secondary development.

4.11.97 How does ESP32 connect to an AP whose RSSI does not fall below the configured threshold when there are multiple APs in the environment?

In ESP32 station mode, there is a wifi_sta_config_t structure with 2 variables underneath, i.e., sort_method and threshold. The RSSI threshold is configured by assigning values to these two variables.

4.11.98 ESP32 Wi-Fi has a beacon lost and sends 5 probe requests to the AP after 6 seconds. If the AP does not respond, disconnection will be caused. Can this 6 seconds be configured?

Use API esp_wifi_set_inactive_time to configure the time.

4.11.99 Does ESP32 Wi-Fi work with PSRAM?

For information on using Wi-Fi with PSRAM, please refer to Using PSRAM.

4.11.100 [Connect] How to troubleshoot the issue that ESP32 series of products cannot connect to the router over Wi-Fi from the hardware and software aspects?

Please follow the steps below to troubleshoot the issue:

- Firstly, use the Wi-Fi error code to determine the possible cause for the failure.
- Then, try connecting another device, such as a phone, to the router to determine whether this is a problem with the router or ESP32.
  - If the phone cannot connect to the router either, please check if there is any problem with the router.
  - If it can, please check whether there is any issue with ESP32.
- Steps to troubleshoot router issues:

  - Check whether the router is in the stage of power off and rebooting. In this stage, the router cannot be connected. Please do not connect to it until it is initialized.
  - Check whether the configured SSID and PASSWORD are consistent with those of the router.
  - Check whether the router can be connected after being configured in OPEN mode.
  - Check whether the router can connect to other routers.
- Steps to troubleshoot ESP32 issues:
  - Troubleshoot the ESP32 hardware:
    - Check whether the issue occurs only in a specific ESP32. If it occurs in a small number of specific ESP32 devices, identify how likely the issue is to occur and compare the hardware differences between them and regular ESP32 devices.
  - Troubleshoot the ESP32 software:
    - Check whether the Wi-Fi connection works using the station example in ESP-IDF. The example has a reconnecting mechanism by default, so please watch if ESP32 can connect to Wi-Fi as it is trying reconnecting.
    - Check whether the configured SSID and PASSWORD are consistent with those of the router.
    - Check whether ESP32 can connect to the router when the router is configured in OPEN mode.
    - Check whether ESP32 can connect to Wi-Fi after calling the API `esp_wifi_set_ps(WIFI_PS_NONE)` additionally before executing the code for connecting to Wi-Fi.
- If all the above steps still fail to locate the issue, please capture Wi-Fi packets for further analysis by referring to Espressif Wireshark User Guide.

4.11.101 After being connected to the router, ESP32 prints `W (798209) wifi:<ba-add>idx:0 (ifx:0, f0:2f:74:9b:20:78), tid:0, ssn:154, winSize:64` and `W (798216) wifi:<ba-del>idx` several times every 5 minutes and consumes much more power. Why?

- This log does not indicate any issue. It is related to the Wi-Fi block acknowledgment mechanism. `ba-add` means the ESP32 received an add block acknowledgment request frame from the router. `ba-del` means the ESP32 received a delete block acknowledgment request frame from the router. Frequent printing of this log suggests that the router has been sending packets.
- If this log is printed periodically every five minutes, it may indicate that the router is updating the group secret key. You could double-check it according to the following steps:
  - Print log in wpa_supplicant_process_1_of_2() to check if this function is called every 5 minutes when the group key is updated every 5 minutes.
  - In the router' s Wi-Fi configuration interface, check if there is the `Group Key Update Time` option and it is set to 5 minutes.

4.11.102 Why can' t ESP32 keep the Wi-Fi sending rate at a fixed value with the function esp_wifi_config_80211_tx_rate() to maintain stable transmission?

- esp_wifi_config_80211_tx_rate() is used to configure the sending rate of esp_wifi_80211_tx().
- To set and fix the Wi-Fi sending rate, use the function esp_wifi_internal_set_fix_rate.

4.11.103 How do I debug the ESP32 station that is connected to a router but does not get an IP properly?

- Open the debug log of DHCP in lwIP, go to ESP-IDF menuconfig, and configure `Component config` > `LWIP` > `Enable LWIP Debug(Y)` and `Component config -> LWIP` > `Enable DHCP debug messages(Y)`.

- Earlier IDF versions do not have the above options, so please refer to lwipopts.h line 806 and 807, to change `LWIP_DBG_OFF` to `LWIP_DBG_ON` in both lines of code as follows.

```
#define DHCP_DEBUG LWIP_DBG_ON
#define LWIP_DEBUG LWIP_DBG_ON
```

4.11.104 When ESP32 works as a softAP, the station connected to it does not get the IP. How to debug?

To open the debug log of DHCP in lwIP for debugging, please go to dhcpserver.c and change `#define DHCPS_DEBUG 0` to `#define DHCPS_DEBUG 1`.

4.11.105 In ESP-IDF menuconfig, after `Component config > PHY > Max Wi-Fi TX power(dBm)` is configured to adjust the Wi-Fi transmit power, what is the actual power? For example, what is the actual maximum transmit power when the option is configured to 17 dBm?

- For ESP32, the actual maximum transmit power in the example is 16 dBm. For the mapping rules, please refer to the function esp_wifi_set_max_tx_power().
- For ESP32-C3, the maximum transmit power value configured in menuconfig is the actual maximum power value.

4.11.106 ESP-IDF currently supports connecting to Chinese SSID routers with UTF-8 encoding. Is there a way to connect to Chinese SSID routers with GB2312 encoding?

Yes, please keep the encoding method of the ESP device consistent with that of the router. In this case, make the ESP device also use the GB2312-based Chinese SSID.

4.11.107 After connecting to the router, ESP32 consumes much power in an idle state, with an average current of about 60 mA. How to troubleshoot the issue?

- Please capture Wi-Fi packets for further analysis. See espressif Wireshark User Guide. After the packets are captured, check whether the NULL data packet sent by the device contains `NULL(1)`. If `NULL(1)` is sent every 10 seconds, it means that ESP32 is interacting with the router in keepalive mode.
- You can also check the `TIM(Traffic Indication Map)` field of the beacon packet in the captured packets. If `Traffic Indication` is equal to 1, it means Group Frames Buffered. In this case, ESP32 will turn on RF, resulting in higher power consumption.

4.11.108 How to configure the Wi-Fi country code when the ESP end product needs to be sold worldwide?

- Different Wi-Fi country codes need to be set for different countries.
- The default country code configuration can be used for most countries, but it is not compatible with some special cases. The default country code is `CHINA {.cc="CN", .schan=1, .nchan=13, policy=WIFI_COUNTRY_POLICY_AUTO}`. While in the ESP-IDF v5.0 or later version, the default one has been changed to `"01" (world safe mode) {.cc="01", .schan=1, .nchan=11, .policy=WIFI_COUNTRY_POLICY_AUTO}`). Since channels 12 and 13 are passively scanned by default, they do not violate the regulations of most countries. Besides, the country code of the ESP product is automatically adapted to the router that it is connected to. When disconnected from the router, it automatically goes back to the default country code.

Note:

- There is a potential issue. If the router hides the SSID and is on channel 12 or 13, the ESP end product can not scan the router. In this case, you need to set `policy=WIFI_COUNTRY_POLICY_MANUAL` to enable ESP end products to actively scan on channels 12 and 13.
- Some countries, such as Japan, support channels 1-14, and channel 14 only supports 802.11b. ESP end products cannot connect to routers on channel 14 by default.

4.11.109 Sometimes the rate drops or even a disconnection occurs after a period of iperf testing. What is the reason and how to solve it?

- Possible reasons:
  - Bad network environment.
  - Incompatibility between the computer or mobile phone and the ESP32-S2 or ESP32-S3 softAP.
  - Modem-sleep affects throughput. In earlier IDF version before v5.1, it is recommended to manually disable Modem-sleep when conducting iperf testing.
- Solutions:
  - In the case of a bad network environment, change the network environment or test in a shielded box.
  - In the case of incompatibility, disable `menuconfig > Component config > Wi-Fi > WiFi AMPDU RX`. If disconnections occur again, disable `menuconfig > Component config > Wi-Fi > WiFi AMPDU TX`.
  - In the case of Modem-sleep, disable Modem-sleep by `esp_wifi_set_ps(WIFI_PS_NONE)`.

Note:

- AMPDU stands for Aggregated MAC Protocol Data Unit and is a technique used in the IEEE 802.11n standard to increase network throughput.
- When `WiFi AMPDU RX` is disabled, the device will not receive AMPDU packets, which will affect the RX performance of the device.
- When `WiFi AMPDU TX` is disabled, the device will not send AMPDU packets, which will affect the TX performance of the device.

4.11.110 Why is this log frequently printed when the phone connects to the ESP32-S3 that works as the Wi-Fi AP based on the ESP-IDF v5.0 SDK?

```
 (13964) wifi:<ba-del>idx
 (13964) wifi:<ba-add>idx:2 (ifx:1, 48:2c:a0:7b:4e:ba), tid:0, ssn:5,␣
↪winSize:64
```

This is because A-MPDU is created and deleted all the time. The printing is only auxiliary and does not affect communication. If you need to hide this log, add the following code before the Wi-Fi initialization code.

```
esp_log_level_set("wifi", ESP_LOG_ERROR);
```

4.11.111 Does ESP32 support the coexistence of the network port (LAN8720) and Wi-Fi (Wifi-AP)?

Yes, this can be achieved by writing the detection events of both connections as one.

4.11.112 How can I optimize ESP32's slow IP address acquisition after Wi-Fi is connected in a weak network environment or interference environment?

- You can disable Modem-sleep using *esp_wifi_set_ps(WIFI_PS_NONE);* after Wi-Fi start, and enable Modem-sleep after getting the event *IP_EVENT_STA_GOT_IP*.
- For the situation of reconnection after disconnection, you can manually disable Modem-sleep before connection, and enable it after getting the event *IP_EVENT_STA_GOT_IP*.
- Note: This optimization is not applicable for Wi-Fi/BT coexistence scenarios.

4.11.113 When ESP32/ESP32-S2/ESP32-S3 series chips work in SoftAP mode, they are susceptible to disconnect from mobile phones and PCs of other manufacturers when they are communicating with each other. How can I optimize this situation?

- It is recommended to turn off `WiFi AMPDU RX` and `WiFi AMPDU TX` options in menuconfig.
- You can further confirm the cause by packet capture.

4.11.114 What is the value range of ESP32 Wi-Fi TX power?

- It ranges from 2 to 20 dBm. In ESP-IDF, you can use `esp_wifi_set_max_tx_power()` to set the maximum of TX power, and use `esp_wifi_get_max_tx_power()` to the get the maximal TX power supported by the system.
- It should be noted that setting TX power too high may affect system stability and battery life, and may also violate wireless regulations in some countries and regions, so it should be used with caution. For more details, please refer to esp_wifi_set_max_tx_power() API.

4.11.115 How do I get Wi-Fi RSSI when using ESP32?

When using ESP32 as a station in ESP-IDF release/v4.1, you can use the following code example to obtain the RSSI of the connected AP:

```
wifi_ap_record_t ap_info;
if (esp_wifi_sta_get_ap_info(&ap_info) == ESP_OK) {
  int rssi = ap_info.rssi;
  // handle rssi
}
```

The `wifi_ap_record_t` structure contains information about the connected AP, including SSID, BSSID, channel, and encryption type. The RSSI field represents the RSSI value of the AP. Call `esp_wifi_sta_get_ap_info()` to get the information of this structure. For details of the API, please refer to esp_err_t esp_wifi_sta_get_ap_info(wifi_ap_record_t *ap_info).

4.11.116 Does ESP32 support WPA3-Enterprise?

- ESP32 supports WPA/WPA2/WPA3/WPA2-Enterprise/WPA3-Enterprise/WAPI/WPS and DPP. For more details, please refer to ESP32 Wi-Fi Feature List.
- esp-idf release/v5.0 has provided wifi_enterprise example, which supports setting WPA3-Enterprise mode for testing. You can configure it as following: `idf.py menuconfig` > `Example Configuration` > `Enterprise configuration to be used` > `WPA3_ENT`.

4.11.117 Does ESP modules support WAPI (Wireless LAN Authentication and Privacy Infrastructure)?

Supported, please refer to the Wi-Fi Feature List.

4.11.118 How can I increase the time for scanning routers when using ESP32 as the Wi-Fi Station to connect routers?

- In ESP32, by default, channels 1 ~ 11 are active scanning, while channels 12 ~ 13 are passive scanning. Active and passive scanning require different amounts of time. For more details, please refer to Wi-Fi scan configuration. The default duration for active scanning is 120 ms per channel, while for passive scanning, it is 360 ms per channel. You can call the following function before calling `esp_wifi_start()` to increase the time for scanning routers:

```
extern void scan_set_act_duration(uint32_t min, uint32_t max);
extern void scan_set_pas_duration(uint32_t time);
scan_set_act_duration(50, 500);
scan_set_pas_duration(500);
```

- Or you can modify the active scanning time and passive scanning time by changing related parameters in the structure `wifi_ap_record_t`.

Note:

- A typical beacon interval is 102.4 ms, so it is recommended to set the active scanning time bigger than this value, which should be at least 120 ms.

4.11.119 Does ESP32 support LDPC?

- Yes. No additional configuration or calling is required as it is already implemented in the driver.
- ESP32-C6 or ESP32-C5 does not support LDPC.

4.11.120 Does the ESP module support WAPI AS?

Not supported. WAPI has two authentication methods, i.e., certificate authentication and pre-shared key authentication. Currently, the ESP module only supports pre-shared key authentication (WAPI-PSK), while WAPI AS is used for certificate authentication.

4.11.121 When encountering the error message `Error occurred during sending: errno 12` during UDP transmission testing, what should I do to resolve it?

- errno 12 indicates insufficient space. This issue occurs when there is a mismatch in processing speed between the upper and lower layers. The upper-layer application continuously generates a large amount of data and sends it to the UDP protocol for transmission. However, the underlying network or the receiving end cannot keep up with the sending speed, resulting in insufficient space.
- To address this problem, you can handle this return value by implementing appropriate measures. For example, upon receiving this return value, the upper layer can initiate a resend mechanism or introduce a delay to slow down the sending speed.

4.11.122 Does the ESP module support Wi-Fi HaLow functionality?

No, Wi-Fi HaLow is based on the 802.11ah protocol.

4.11.123 How to prevent any radio frequency waves from being sent in ESP32 Wi-Fi Scan mode?

- Use the esp_wifi_scan_start() API to set ESP32 to passive scan mode WIFI_SCAN_TYPE_PASSIVE.

4.11.124 What is the default 802.11 Wi-Fi protocol when the ESP32-C6 is in Wi-Fi AP mode?

- When ESP32-C6 enables Wi-Fi AP mode, it uses the mixed mode of 802.11b/g/n by default. The protocol can be set via esp_wifi_set_protocol().
- ESP32-C6 does not support 802.11ax in AP mode.

4.11.125 Why can't the ESP32 Wi-Fi Station connect to a Wi-Fi hotspot in 2.4 GHz Enhanced Open mode?

- Connection to 2.4 GHz Wi-Fi hotspots in Enhanced Open mode is only supported on ESP-IDF release/v5.2 and above. Additionally, the `Component config > ``Wi-Fi > Enable OWE STA` configuration option should be enabled. For more information, please refer to the Wi-Fi Enhanced Open guide.
- When connecting to a router in this security mode, do not set any password.

4.11.126 How to determine whether the connected Wi-Fi applies Wi-Fi 4 or Wi-Fi 6 ?

- You can use esp_wifi_sta_get_negotiated_phymode(wifi_phy_mode_t *phymode) to obtian the mode of the current connected station. Here is an example:

```
wifi_phy_mode_t phymode;
esp_wifi_sta_get_negotiated_phymode(&phymode);
printf("111=%d\n",phymode);
```

- If the printed value is 3, it indicates the connection with the station applies Wi-Fi 4. If the printed value is 5, it indicates the connection is based on Wi-Fi 6.

4.11.127 Does ESP32-S3 support AP and STA working simultaneously?

- Yes. Please refer to the example softap_sta.

4.11.128 Does the ESP Wi-Fi module support power save mode in SoftAp mode?

This feature is currently not supported.

4.11.129 What is the maximum ranging bandwidth supported by ESP32S3 FTM?

ESP32-S3 FTM supports maximum ranging bandwidth up to 40 MHz.

4.11.130 Do ESP chips support one STA conducting FTM with multiple APs at the same time?

Not supported. STA can only perform FTM with one AP at a time.

4.11.131 In WiFi Station mode, which WiFi mode will ESP32-C6 use to establish a connection with the router when both WiFi 4 and WiFi 6 modes are enabled on the router?

ESP32-C6 will prioritize establishing a connection with the router using WiFi 6.

4.11.132 In WiFi AP mode, how many WiFi Station devices can ESP32-C2 series products support at most?

- Due to hardware limitations, ESP32-C2 can support up to four WiFi Station connections in WiFi AP mode.
- The maximum number of connections supported in WiFi AP mode varies between different chips. For details, please refer to esp-idf/components/esp_wifi/include/esp_wifi_types.h.

4.11.133 Does ESP32 support WPA3 WiFi AP mode?

Yes. ESP-IDF v5.1 and above support WPA3 WiFi AP mode. Please refer to the description in ESP-IDF v5.1 Release.

4.11.134 Do ESP chips support Short GI in SoftAP mode?

This feature is currently not supported.

4.11.135 Do STA and AP need to establish a connection first before executing FTM?

No connection establishment is required, and the STA can directly initiate FTM through the AP. It is recommended to scan the AP first to confirm whether it supports FTM Responder mode. For specific operations, you can refer to the FTM example in the esp-idf/wifi directory.

4.11.136 Do ESP chips support uAPSD in SoftAP mode?

uAPSD is a Wi-Fi capability that provides more power consumption savings on the client, in low periodic latency-sensitive traffic modes (such as VoIP). ESP chips currently do not support this feature.

4.11.137 When enabling WiFi AP mode on products based on Espressif series, is it possible to set up automatic channels?

Espressif series products use the WiFi SoftAP mode, which does not support automatic channel settings.

4.11.138 Do ESP series products support roaming in Wi-Fi AP mode?

- No, ESP series products only support connecting to routers with roaming features in Wi-Fi Station mode.
- Software Reference: esp-idf/examples/wifi/roaming.

4.11.139 Does the router's DHCP lease duration affect the module's Wi-Fi connection?

The DHCP protocol handles lease renewal automatically, without requiring a device or Wi-Fi restart. However, if the renewal fails, the module may lose its IP address. In this case, The IP address needs to be detected and acquired by the software again.

4.11.140 If the module's authentication mode is set to WIFI_AUTH_OPEN, but the router uses a higher-security authentication mode like WIFI_AUTH_WPA3_PSK, can the module still connect?

Yes, the module can connect to the router successfully. The minimum authentication mode (wifi_sta_config_t threshold.authmode) means that the router's encryption level must be no lower than this value. Since IDF v5.0, authmode must be set to WIFI_AUTH_WEP or WIFI_AUTH_WPA_PSK if you need to connect to a WEP or WPA network.

4.11.141 How to handle the issue "esp-aes: Failed to allocate memory" ?

This issue indicates a failure in DMA memory allocation. You can print the DMA memory status to check if it is sufficient. If the memory is insufficient, try to optimize the code to reduce DMA memory usage by other functions or interfaces.

4.11.142 How to completely disable Wi-Fi configurations on ESP32 to optimize firmware size when Wi-Fi features are not required?

- In terms of software, simply avoid calling esp_wifi_init(). This ensures no Wi-Fi code and configurations is included, having no impact on the firmware size.
- For further optimization, refer to Minimizing Binary Size.

4.11.143 Why can't a router with WPA2 encryption be connected when the WiFi scan auth mode threshold is set to WPA2_WPA3_PSK in menuconfig, but only routers with WPA3 or WPA2/WPA3 encryption can?

- This is due to the ordering of the authmode threshold: `OPEN < WEP < WPA_PSK < OWE < WPA2_PSK = WPA_WPA2_PSK < WAPI_PSK < WPA3_PSK = WPA2_WPA3_PSK = DPP`. In other words, WPA2_WPA3_PSK uses WPA3_PSK as the authmode threshold, so it cannot connect to a router that only supports WPA2 encryption.

4.11.144 What is the coexistence relationship between ESP32-C6's Wi-Fi 6 mode and HT40?

By default, ESP32-C6 connects using the latest protocol, i.e., 802.11ax (Wi-Fi 6). However, Wi-Fi 6 mode of ESP32-C6 only supports 20 MHz bandwidth. If the router is set to HT40 bandwidth, or is forced to 40 MHz bandwidth using *esp_wifi_set_bandwidth(0, WIFI_BW_HT40)* in the program, ESP32-C6 will switch to the 802.11bgn protocol instead of 802.11ax (Wi-Fi 6).

4.11.145 When running the ESP8266_RTOS_SDK/examples/wifi/getting_started/station example on ESP8266, it cannot connect to the esp-idf/examples/wifi/getting_started/softAP example on ESP32-S3 with esp-idf v5.3 as Wi-Fi AP mode hotspot. However, it can connect to the esp-idf/examples/wifi/getting_started/softAP example on ESP32-S3 with esp-idf v5.0 as Wi-Fi AP mode hotspot. What is the reason?

- In the `esp-idf/examples/wifi/getting_started/softAP` example of esp-idf v5.3, Wi-Fi PMF (Protected Management Frames) feature is enabled by default, i.e., .pmf_cfg = { .required = true, }, . This requires the peer Station device to support Wi-Fi PMF function. However, ESP8266 does not support PMF, so it cannot connect to the Wi-Fi AP hotspot of esp-idf v5.3.
- But in the `esp-idf/examples/wifi/getting_started/softAP` example of esp-idf v5.0, the default setting is .pmf_cfg = { .required = false, }, , that is, it does not forcibly require Station devices to support Wi-Fi PMF, so it can connect to such Wi-Fi AP hotspots.
- Therefore, when the station device does not support Wi-Fi PMF function, and it needs to connect to a SoftAP, you can modify the setting as `.pmf_cfg = { .required = false, },`.

4.11.146 Does ESP32-C5 support simultaneous use of 2.4GHz and 5GHz Wi-Fi?

- ESP32-C5 chip supports two antenna interfaces for 2.4GHz and 5GHz, but only one can be used at a time. Antenna selection is controlled via software by switching GPIO levels. For implementation details, refer to the esp-idf/examples/phy/antenna example.
- Modules of ESP32-C5 series (such as ESP32-C5-WROOM-1) provide only a single antenna interface, so there is no antenna switching physically speaking. In this case, the same antenna is used for both frequency bands (2.4GHz and 5GHz), with software controlling the switching through time-division multiplexing. Users cannot manually switch antennas via GPIO.

4.11.147 When using an ESP32 device, can I enable smartconfig provisioning (esp_smartconfig_start) while maintaining an existing Wi-Fi connection, so as to switch to another Wi-Fi at any time?

This is not supported. When smartconfig provisioning (esp_smartconfig_start) is enabled, the device enters promiscuous mode and switches channels, causing the existing Wi-Fi connection to be disconnected.

4.11.148 After successful Wi-Fi provisioning with the Blufi example, why does the chip automatically reconnect to the previously configured Wi-Fi hotspot after reset and restart?

The software enables Wi-Fi NVS flash provisioning by default, which saves the Wi-Fi NVS provisioning information in the NVS partition of the flash. After the chip is reset and restarted, it directly reads the Wi-Fi information in the NVS partition for automatic reconnection. Please refer to the Wi-Fi Configuration Phase for details.

4.11.149 Why doesn't the AP receive an ACK when the Wi-Fi STA sends data?

If the AP does not receive an ACK from the STA, it may be due to a Wi-Fi signal problem, or the STA's TX buffer is full. It is recommended to check the Wi-Fi signal strength, and ensure the STA's TX buffer has sufficient space. In addition, the problem can be further analyzed by capturing the Wi-Fi air interface packets.

4.11.150 Why does the power consumption increase when `wifi:bcn_timeout, ap_probe_send_start` appears in the Wi-Fi STA log?

- Network environment issues: The network environment may be poor, or the router may have experienced a power failure. In such cases, the STA device might not receive the beacon for a long time, extending the time to receive packets, which leads to increased power consumption;
- Active AP Detection: When a `bcn_timeout` occurs, the STA attempts to send up to 5 Probe Requests. If the AP sends a Probe Response, the connection is maintained; if the AP does not respond, the STA will send a Disconnect event and disconnect. During this process, the additional packet transmissions can also lead to increased power consumption.

4.11.151 Can ESP32-C5 sniff Wi-Fi 5 and Wi-Fi 6 packets at 20 MHz?

Yes, ESP32-C5 can sniff Wi-Fi 5 and Wi-Fi 6 packets at 20 MHz, provided that these packets are transmitted on the same channel. If the packets are on different channels, they cannot be sniffed.

4.11.152 Why does the ESP32 timeout when pinging a public IP through a 4G dial-up connection?

Possible reasons:

- Network fluctuations from the carrier. It is recommended to test with a different network to verify the problem.
- AP mode may affect forwarding. If ESP32 is in AP mode with terminal device connected, the default routing may switch from the 4G network to the AP interface. It is recommended to disable AP mode for verification.

4.11.153 Within a local area network, is it possible to find the MAC address of a mobile phone through UDP data content without using ARP?

It is not recommended to attempt this. Point-to-point communication requires knowledge of the peer device's MAC address, which is exactly the purpose of ARP (for IPv4) and NDP (for IPv6 via ICMPv6). Any custom method you implement will not be supported on the mobile device side.

4.11.154 Does the Wi-Fi of ESP32-C5 support the simultaneous use of sniffer (promiscuous mode) and STA mode?

Yes, RF coexistence is supported. Sniffer mode and STA mode can be configured to operate in parallel via the Wi-Fi API *esp_wifi_set_promiscuous(true)*.

4.11.155 When enabling the 5G band softAP on C5, will DFS be activated to avoid interference with radar on the same channel?

The current C5 hardware cannot detect radar signals and therefore cannot temporarily disable DFS channels. However, logic to disable DFS channels will be considered in future updates.

4.11.156 How to configure the working frequency band (2.4 GHz/5.8 GHz) of ESP32-C5?

The ESP32-C5 does not currently support simultaneous dual-band operation; it can operate on only one band at a time. You can use esp_wifi_set_band_mode() to set the band mode. The default value is WIFI_BAND_MODE_AUTO, which scans both 2.4G and 5G bands, allowing flexible selection. In test environments where multiple hotspots with the same SSID exist on both bands, the device connects to the one with the strongest RSSI. When set to WIFI_BAND_MODE_5G_ONLY, it can only connect to 5G hotspots. When set to WIFI_BAND_MODE_2G_ONLY, it can only connect to 2.4G hotspots. For dual-band unified SSIDs, you can adjust the preference for 2.4G or 5G using the rssi_5g_adjustment parameter.

4.11.157 Does Espressif chips support the ARP aging mechanism? What is the aging time, and can it be modified? For example, when connecting via Wi-Fi, the MAC address of the AP is obtained through ARP. How long is the interval before the next ARP request is sent?

- ARP aging mechanism is supported.
- The default aging time is 5 minutes, defined by *ARP_MAXAGE*. Currently, no API or menuconfig option is available to modify this value.
- When accessing an IP address, the chip will first checks the ARP table. If there is no entry in the ARP table or the entry is about to expire, it will send an ARP request. After receiving the ARP response, the chip will update the ARP table and keep the record valid for approximately 5 minutes.

4.11.158 What does `LWIP_GARP_TMR_INTERVAL` do? What is the interval in seconds for sending ARP?

GARP is a mechanism where an Espressif chip proactively sends its IP and MAC address mapping, which is used to notify other devices to update their ARP tables. INTERVAL represents the sending interval defaulted to 60 seconds.

4.11.159 After disabling modem sleep, the ping latency is still high. How can it be further optimized?

- In an open wireless environment, Wi-Fi latency is affected by channel competition and air interface fluctuations, and the optimization effect at the lwIP level is limited.
- The following optimizations can be performed:
  1. Try to disable the WIFI AMPDU feature;
  2. Refer to the relevant documents in Wi-Fi Driver based on your target, and try to improve Wi-Fi performance.
- If the peer is also an ESP device and requires extremely low latency, it is recommended to use direct connection protocols such as ESP-NOW to reduce latency.

4.11.160 When using a packet capture tool to analyze Wi-Fi traffic, is it normal that the data remains encrypted even after entering the SSID and password?

Yes, this is normal.

- For WPA2-encrypted networks, in addition to the network password, the packet capture must include the complete four-way EAPOL handshake (EAPOL 1–4). If the full handshake is not captured, the packet capture tool will be unable to decrypt any subsequent data traffic. In this case, the device must reconnect to the network and packets must be captured again to obtain the complete handshake.
- For WPA3-encrypted networks, the network password and a complete handshake alone are not sufficient. The PMK (Pairwise Master Key) is also required for decryption. The PMK can be obtained from printout logs by enabling `ESP_WIFI_DEBUG_PRINT`.

4.11.161 Does ESP32-C5 support AP mode on the 5GHz band? Do 2.4 GHz and 5 GHz share the same MAC address in AP mode?

- ESP32-C5 supports working in AP mode on the 5GHz band. It supports HT40 bandwidth under IEEE 802.11n. However, 40 MHz bandwidth is not supported under IEEE 802.11ac or IEEE 802.11ax. In 802.11ax mode, only HE rates and BSS Color are supported; other features such as OFDMA and MU-MIMO are not supported. In addition, C5 currently does not support simultaneous dual-band operation, and can only work on a single band at any given time.
- In AP mode, 2.4 GHz and 5 GHz share the same MAC address.

4.12 Zigbee

4.12.1 Does the Zigbee protocol stack of ESP32-H2 and ESP32-C6 support automatic frequency hopping under interference?

The Zigbee protocol does not mandate automatic frequency hopping. However, this functionality can be implemented at the application layer: by periodically monitoring channel quality, the system can detect severe interference and coordinate a network-wide migration to a different channel.

Chapter 5

Hardware related

5.1 Chip Comparison

5.1.1 What is the difference between ESP32 with a single core and ESP32 with dual cores regarding programming method, feature performance, power consumption, and so on?

Compared with the single-core chip, the dual-core chip has one more independent core, on which some highly real-time operations can be supported in a better way.

- The two chips employ the same programming method except the following step only. For single-core chip, you have to configure FreeRTOS to make it run on the first core. The configuration path is `make menuconfig` > `Component config` > `FreeRTOS` > `[*] Run FreeRTOS only on first core`. This step is not needed for the dual-core chip.
- The two chips have similar performance in most cases, except under high-load conditions, such as AI algorithm and high real-time interrupts.
- The dual-core chip consumes a little more power than the single-core chip in Modem-sleep mode. For more details, please refer to ESP32 Series Datasheet.

5.1.2 What's the difference between ESP32 v3.0 and previous chip revisions in software and hardware?

- For software usage, there is no difference. ESP32 v3.0 is compatible with old firmwares. For hardware, some bugs have been fixed in ESP32 v3.0.
- For more information on design changes, please refer to ESP32 Chip Revision v3.0 User Guide.

5.1.3 Can the GPIO34 ~ GPIO39 of ESP32 only be set to input mode?

- Yes, the GPIO34 ~ GPIO39 of ESP32 can only be set to input mode. They cannot be set to the output mode as they cannot be pulled up or down by software.
- For more details, please refer to GPIO & RTC GPIO.

5.1.4 Does ESP32 support a driver for Linux?

You can use the esp-hosted solution. This solution currently offers three main variants: **ESP-Hosted-NG**, **ESP-Hosted-FG**, and **ESP-Hosted-MCU**. For differences between the variants, see Variant Comparison.

- Both **ESP-Hosted-NG** and **ESP-Hosted-FG** support Linux hosts:
  - **ESP-Hosted-NG** provides a standard 802.11 Wi-Fi interface, and supports both STA and SoftAP modes.
  - **ESP-Hosted-FG** provides an 802.3 Ethernet interface, and supports STA, SoftAP, and STA+SoftAP modes.
- If you use **ESP-Hosted-FG** and the target Linux platform is not yet adapted, you can refer to the ESP-Hosted-FG Porting Guide.
- If you use **ESP-Hosted-NG** and the target Linux platform is not yet adapted, you can refer to the ESP-Hosted-NG Porting Guide.

5.1.5 What is the meaning of the acquired data when you scan the data matrix on the module's shielding case?

- Taking 0920118CAAB5D2B7B4 as an example. '09' indicates the factory code; '20' indicates the last two numbers of the year of manufacturing, which is 2020 in this example; '11' indicates the week number of manufacturing; and the last 12 numbers '8CAAB5D2B7B4' is the MAC address of this module. For the latest information, please refer to Module Packaging Information.

5.1.6 Can VDD3P3_RTC of ESP32 support independent battery power supply?

- The internal RTC of ESP32 cannot work independently as it requires the main CPU to participate in the configuration. Even if the RTC domain is powered by an independent battery, it may still be powered down suddenly.
- If you want to save the clock information when power-off occurs, please use an external RTC clock chip.

5.1.7 What is the difference between ESP32-PICO-D4, ESP32-PICO-V3, and ESP32-PICO-V3-02?

- Built-in chip version: The core chip of ESP32-PICO-V3 and ESP32-PICO-V3-02 is ESP32 (ECO V3), while that of ESP32-PICO-D4 is ESP32 (ECO V1).
- Package size: The size of ESP32-PICO-D4 and ESP32-PICO-V3 is 7 × 7 × 0.94 (mm), while the size of ESP32-PICO-V3-02 is 7 × 7 × 1.11 (mm).
- Built-in flash: ESP32-PICO-D4 and ESP32-PICO-V3 integrate 4 MB SPI Flash, while ESP32-PICO-V3-02 integrates 8 MB flash and 2 MB PSRAM.
- Pin differences: Please refer to Section Compatibility with ESP32-PICO-V3 in the ESP32-PICO-V3-02 Datasheet.

5.1.8 Does ESP8266 support 32 MHz crystal?

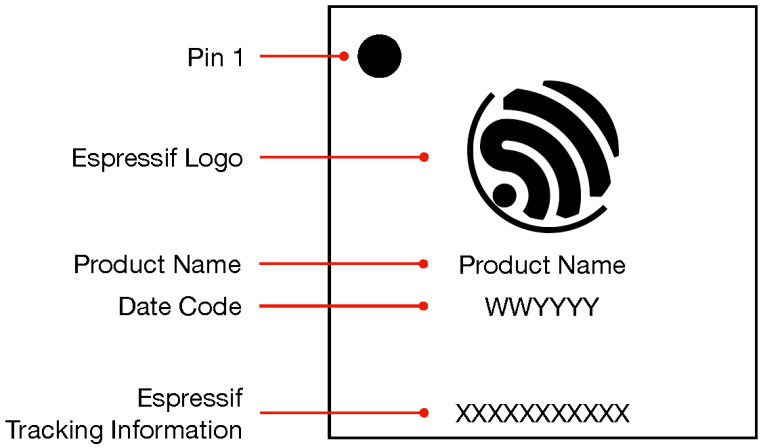
- No. Currently, ESP8266 supports 26 MHz and 40 MHz crystals, and the 26 MHz crystal is recommended.

5.1.9 Do ESP products support Zephyr?

- Please refer to Espressif ' s Support for Zephyr for detailed information about the support for Zephyr in ESP products. While there are only a few functional modules have been adapted so far, other modules will be updated further later. For feature requests, you may check or ask on the Zephyr GitHub issue first.
- You can also find information about ESP products in sections like XTENSA Boards and RISCV Boards in *Zephyr Docs <https://docs.zephyrproject.org/latest/introduction/index.html>*.

5.1.10 How to identify the chip revision from the chip silk marking?

You can do it by checking the second character of Espressif Tracking Information line in the bottom of the chip silk marking.



- For detailed differences between chip revisions, please refer to respective chip errata documents from Espressif ' s documentation page.
- For a complete explanation of the chip silk marking, please refer to Espressif Chip Packaging Information.

5.2 Development board

5.2.1 Does the ESP32-Korvo v1.1 development board integrate an LED driver chip?

Yes.

5.2.2 How to improve the overheating situation of the ESP-EYE development board?

- If the camera is not always on, and Wi-Fi transmits data periodically, the development board can enter sleep mode during idle time to reduce power consumption.
- Add a heat sink on top of the ESP32 chip to increase the heat dissipation area.

5.2.3 If a development board is not powered up through USB, how to use pins to supply power for it?

- First option: Connect the 3V3 pin to the 3V3 pin and GND to GND (If there are components on the board that are not powered by 3.3 V, these components will not work).
- Second option: Connect the 5V pin to the 5V pin and GND to GND.

Note: The power supply current should be bigger than 500 mA.

5.2.4 In ESP32-Korvo-DU1906, how does the DU1906 chip interact with ESP32 for audio data?

In ESP32-Korvo-DU1906, the audio data of DU1906 is transmitted to ESP32 through SPI.

5.2.5 Is there an Ethernet development board that supports POE power supply?

ESP32-Ethernet-Kit can satisfy this requirement.

5.2.6 The LED light on the ESP32-DevKitC development board does not light up and the device manager cannot find the device. Why?

- Check if the power supply is normal: After supplying power for the board through USB, use a multimeter to test whether there is voltage between the VCC and GND pins.
- Check if it is a specific development board fault: Check if other ESP32-DevKitC development boards can be powered up with this USB cable.
- If you cannot locate the reason using the above methods, you can connect the board through a USB to TTL device. You only need to connect the VCC, GND, TXD pins of ESP32-DevKitC to test whether it is caused by chip problems, and to check with the serial port tool whether it can print logs.
- If possible, please test whether the serial port driver chip has voltage. For the circuit, you can refer to the ESP32-DevKitC schematic.

5.2.7 Why can ' t I find the EN button mentioned in the documentation on the development board?

It is recommended to check if the development board has a Reset button. As the EN button is often used for resetting, some development boards will mark it as a Reset button.

5.2.8 I can't find the serial port in the device manager after connecting the ESP32 development board to a Windows computer. What could be the reasons?

After connecting the ESP32 development board to a Windows computer, if you can't find the serial port in the device manager, it could be due to the following reasons:

- Driver not installed: Before connecting the ESP32 development board to a Windows computer, you need to install the driver. If the driver is not installed or installed incorrectly, the development board will not be recognized as a serial port device. Download and install the FT232R USB UART driver.
- Loose or damaged USB cable: If the USB cable is loose or damaged, the development board cannot be correctly recognized. Users can replace the USB cable or check if the USB cable is plugged in tightly to ensure a normal connection between the USB cable and the computer.
- Faulty development board: If the above two reasons can be excluded, it may be because development board itself is faulty. Users can try connecting to other USB ports or other computers for testing, or detect and repair the development board.

It should be noted that when testing the connection of the development board, you need to confirm whether the serial port settings and driver settings of the development board are correct. Some development boards need to manually select the correct port and baud rate in the serial port settings to connect to the computer. At the same time, some drivers also need to manually set the port and baud rate to ensure consistency with the development board settings.

5.2.9 For the ESP32-LyraT v4.3 audio development board, it's hard to enter download mode even when the Boot button is held down for a long time. What could be the reason?

The correct procedure is holding down the Boot button and pressing the RST button (without releasing the Boot button), and then releasing the RST button (with the Boot button still held down). When the board enters download mode and starts downloading, you can release the Boot button.

5.2.10 How long does it take for the ESP-WROOM-02D module to restart after the reset signal?

It will restart when the input level is lower than 0.6 V for more than 200 µs.

5.2.11 According to the schematic of ESP32-LyraT-Mini, the analog output of the ES8311 codec chip is connected to the input of the ES7243 ADC chip. What is the purpose of this?

The hardware acquisition circuit of the AEC reference signal simultaneously transmits the DAC output of the Codec (ES8311) to the speaker PA and the ADC (ES7243) AINLP/N, of which the signal collected would be send back to the ESP32 as the reference signal for the AEC algorithm.

5.2.12 When using the ESP32-MINI-1 module, the serial port printed the follows log when powered on. What could be the reason?

```
rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
invalid header: 0xffffffff
invalid header: 0xffffffff
invalid header: 0xffffffff
invalid header: 0xffffffff
invalid header: 0xffffffff
invalid header: 0xffffffff
invalid header: 0xffffffff
invalid header: 0xffffffff
ets Jul 29 2019 12:21:46
```

The is because flash is not programmed.

5.2.13 Which GPIO is connected to the RGB LED of the ESP32-S3-DevKitC-1 development board?

- The RGB LED on the ESP32-S3-DevKitC-1 v1.0 development board is connected to GPIO48.
- The RGB LED on the ESP32-S3-DevKitC-1 v1.1 development board is connected to GPIO38.
- The reason why the ESP32-S3-DevKitC-1 v1.1 development board changed the RGB LED pin to GPIO38 is that the ESP32-S3R8V chip's VDD_SPI voltage has been set to 1.8 V. Therefore, unlike other GPIOs, GPIO47 and GPIO48 in the VDD_SPI power domain of this chip also operate at 1.8 V.

5.3 Hardware Design

5.3.1 The I2S pins of ESP32 are scattered. Can I route I2S signals to adjacent pins? For example, to `GPIO5`, `GPIO18`, `GPIO23`, `GPIO19`, and `GPIO22`, or to `GPIO25`, `GPIO26`, `GPIO32`, and `GPIO33`.

All I2S signals can be routed to different I/Os freely. Please note that some I/Os can only be set as input. For details, please refer to Section *Peripheral Pin Configurations* and Appendix *IO_MUX* in the ESP32 Datasheet.

5.3.2 How can I stop the power loss of VDD3P3_RTC after ESP32 enters Light-sleep mode?

After entering Light-sleep mode, if RTC power loss occurs, the level of the GPIO corresponding to the VDD3P3_RTC pin will be pulled low, causing external RTC or other devices to malfunction. There are two possible ways to solve this problem:

- Set the RTC hardware voltage control register (RTC_CNTL_REG) to control the voltage. Specifically, set the FORCE_PU and FORCE_PD bits in the RTC_CNTL_REG register to 1, namely, `RTC_CNTL_REG |= RTC_CNTL_FORCE_PU_M | RTC_CNTL_FORCE_PD_M;`.
- Set GPIO hold pins. The ESP32's Light-sleep mode supports GPIO hold function, which can set some GPIO pins as hold pins to maintain their voltage level when the system enters Low-power mode. Specifically, the VDD3P3_RTC pin can be set as a hold pin to maintain its voltage. The relevant code snippet is as follows:

```
esp_sleep_pd_config(ESP_PD_DOMAIN_RTC_PERIPH, ESP_PD_OPTION_ON);
esp_sleep_enable_gpio_wakeup();
gpio_hold_en(GPIO_NUM_X);
```

Among them, ESP_PD_DOMAIN_RTC_PERIPH represents the power domain of the RTC subsystem. ESP_PD_OPTION_ON enables the power domain. And the gpio_hold_en() function can set the specified GPIO pin as a hold pin. After setting the VDD3P3_RTC pin as a hold pin, even if the system enters Light-sleep mode, the voltage of the pin will be held.

Please note that using the GPIO hold function will increase the power consumption of the system, so the appropriate solution should be selected according to the specific application scenario. If only the power supply of the RTC hardware needs to be held, the first method can be adopted. If the power supply of other external devices needs to be held as well, the second method can be adopted.

5.3.3 What should be noted when I configure the pins of ESP32?

- You may assign most of the digital peripherals to any pins through GPIO Matrix. However, functions such as SDIO, high speed SPI, and analog can only be realized via IO MUX.
- For details, please refer to GPIO & RTC GPIO.

Note:

- Strapping pins have default levels. Please refer to ESP32 Datasheet.
- GPIO34 ~ GPIO39 can only be set as input without software-enabled pull-up or pull-down functions.
- GPIO6 ~ GPIO11 are saved for flash.
- GPIO1 and GPIO3 are the TX and RX pins for UART0, which cannot be configured.
- GPIO16 and GPIO17 are saved for PSRAM if there is any.

5.3.4 What is the voltage tolerance of GPIOs of ESP chips?

The voltage tolerance of GPIO is 3.6 V. If the voltage exceeds 3.6 V, please add a voltage divider to protect GPIO pins from damage.

5.3.5 What are the power supply specifications for ESP8266?

- Digital working voltage range: 1.8 V ~ 3.3 V
- Analog working voltage range: 3.0 V ~ 3.6 V (the lowest possible value is 2.7 V)
- Peak analog circuit current: 350 mA
- Peak digital circuit current: 200 mA

Note: The operating voltage of SPI flash should be compatible with that of GPIO pins. The operating voltage of CHIP_EN ranges from 3.0 V to 3.6 V, so please use a level converter when GPIO pins operates at 1.8 V.

5.3.6 Do Espressif Wi-Fi modules support single-layer PCBs?

- The ESP32 module is a wireless device, which needs rather high-quality PCB materials to fulfill the RF performance requirements. We have tested four-layer and two-layer PCBs, but not single-layer ones.
- Single-layer PCBs are not recommended as RF performance cannot be guaranteed. You may use single-layer PCBs in your end products and then mount Espressif modules.
- Four-layer PCBs are recommended for desired RF performance.

5.3.7 What should be noted when I power ESP8266 with batteries?

- The operating voltage of ESP8266 ranges from 3.0 V to 3.6 V, so two AA batteries can be used to power ESP8266. Please ensure the battery voltage stays within the operating range of ESP8266 when it drops.
- If the lithium battery voltage surpasses module operating voltage, and the voltage drops heavily during discharge, then such batteries should not be used to power ESP8266.
- We recommend you to use DC/DC converters or LDO regulators to convert voltage before powering ESP8266. Please pay attention to the difference between the input and output voltages of converters or regulators.

5.3.8 Where can I get the footprint of ESP32 Series?

You may get the footprint in the PCB layout of different modules. Please refer to reference designs.

5.3.9 For ESP32-S2 chips, can I have audio connection when the DVP camera interface is in use?

The LCD, DVP camera, and I2S interfaces of ESP32-S2 share one set of hardware, so they cannot be used at the same time.

5.3.10 What should be noted when I assign I2C signals to GPIO0 and GPIO4 of ESP32 modules?

Please pull GPIO0 up when assigning I2C signals to the pin. Please also ensure GPIO0 can be pulled down when powered on during flashing, which can be released afterwards. Only pull GPIO0 down when flashing firmware on ESP32 modules.

5.3.11 When the external flash is connected to GPIO6 ~ GPIO11, can they be set as SPI pins?

When the external flash is connected to GPIO6 ~ GPIO11, they cannot be set as SPI pins.

5.3.12 Do I need to connect an external crystal when using the ESP8285 chip?

You need to connect an external crystal, as the chip has no internal crystal.

5.3.13 Where can I find the reference design for connecting an external PSRAM to ESP32-D2WD?

You may refer to the design for the external PSRAM of ESP32-PICO-D4. Please refer to Chapter *Peripheral Schematics* in the ESP32-PICO-D4 Datasheet.

Note: ESP32-D2WD has an 1.8 V flash, so please add a resistor and a capacitor to VDD_SDIO and connect an 1.8 V PSRAM.

5.3.14 Can I use ESP32 to play music with PWM or DAC?

You may use ESP32 to play music with PWM or DAC, and we recommend you play voice prompts. To run a test, please refer to esp-adf/examples/player/pipeline_play_mp3_with_dac_or_pwm.

5.3.15 Why is the suggested voltage range of ESP32 modules different from that of ESP32 chips?

- This is because of the different working environments and usage scenarios.
  - The ESP32 chip is a bare die and requires external circuitry on a circuit board to function properly. The recommended operating voltage range for the ESP32 chip is 2.3 V to 3.6 V, which is determined by the chip's electrical parameters. Within this voltage range, the ESP32 chip can function properly and provide optimal performance and power consumption.
  - The ESP32 module, on the other hand, is a packaged electronic module that typically includes voltage regulators, external crystals, external antennas, and other peripheral chips, such as flash memory and RAM, and can be used directly. As the module's circuitry has already been optimized and tested, its recommended operating voltage range may be narrower. For example, the ESP32-WROOM-32 module has a recommended operating voltage range of 3.0 V to 3.6 V. Apart from that, as the module has to take flash voltage into account, the recommended operating voltage for the ESP32 module would thus be higher.
- When using these chips and modules, appropriate power supplies and peripheral circuits should be chosen based on the specific situation to ensure that they function properly.
- For more information, please check module and chip datasheets.

5.3.16 Why does it take a longer time to erase the flash of self-developed modules than that of Espressif modules?

- It is common that the erasing time vaires, as it depends on factors such as the manufacturer of your flash and the size of the block you erase.
- If you want to shorten the erasing time, you may test flash memories from different manufacturers.

5.3.17 Why does the current surge when ESP8266 is powered on?

- The RF and digital circuits of ESP8266 are highly integrated. When ESP8266 is powered on, the RF automatic calibration starts to work, which requires high current.
- The maximal current of the analog circuit can reach 500 mA, while that of the digital circuit is 200 mA.
- Usually the average current is 100 mA.
- To wrap up, ESP8266 needs a 500 mA power supply.

5.3.18 What choices do I have when configuring the RMII clock for the Ethernet of ESP32?

- We recommend use GPIO0 as the RMII clock input pin. Please note that the GPIO0 cannot be low level when the chip powered on.
- For details, please refer to the Configure MAC and PHY guide.

5.3.19 What kind of socket is used on ESP32-LyraT development boards to connect a speaker?

Please use a PH-2A socket.

5.3.20 For modules housing ESP32, which pins cannot be set by users?

- For ESP32-WROOM Series of modules, GPIO6 ~ GPIO11 are pins for flash and cannot be set for other uses.
- For ESP32-WROVER Series of modules, GPIO16 and GPIO17 are pins for PSRAM and cannot be set for other uses.
- Besides, please note that ESP32 has five strapping pins. For details, please refer to ESP32 Datasheet.

5.3.21 Which is the reset pin of ESP32?

CHIP_PU serves as the reset pin of ESP32. The input level (VIL_nRST) for resetting the chip should be low enough and remain so for a period of time. Please refer to Section *Reset* in the ESP32 Hardware Design Guidelines.

5.3.22 What should be noted when I design the power supply for ESP8266?

- If you use LDO regulators, please ensure the input voltage ranges from 2.7 V to 3.6 V and the output current is greater than 500 mA.
- The decoupling capacitor must be as close to the chip as possible. The equivalent resistance should be low enough.
- ESP8266 is not 5 V tolerant. It operates at 3.3 V, with the operating voltage ranging from 2.7 V to 3.6 V.
- If you use DC/DC converters, please add LC filters when necessary.
- Please refer to Section *Power Supply* in the ESP8266 Hardware Design Guidelines.

5.3.23 When I use the TOUT pin of ESP8266 to collect ADC sample signals, will the pins be damaged if the voltage is greater than 1.0 V?

- If the input voltage is within the operating range of pins (0 V ~ 3.6 V), the pins will not be damaged.
- If the voltage is greater than 1.0 V, it may lead to abnormal results.

5.3.24 For modules with PCB antennas, what should be noted when I design the PCB and the housing of the antenna?

- When adopting on-board design, you should pay attention to the layout of the module on the base board. The interference of the base board on the module's antenna performance should be reduced as much as possible.
- It is recommended that the PCB antenna area of the module be placed outside the base board, while the module be put as close as possible to the edge of the base board so that the feed point of the antenna is closest to the board.
- Please make sure that the module is not covered by any metal shell. The antenna area of the module and the area 15 mm outside the antenna should be kept clean (namely no copper, routing, components on it).
- For details, please refer to Hardware Design Guidelines.

5.3.25 Can GPIO 34 ~ GPIO39 of ESP32 be used as UART RX pins?

GPIO 34 ~ GPIO39 can be used as UART RX pins.

5.3.26 Where can I find the design reference for the external 32 kHz crystal of ESP32 modules?

Please refer to the hardware design instructions in the document RTC Clock Source (Optional) (ESP32).

5.3.27 Does the flash of ESP32 modules support 80 MHz QIO access mode?

- ESP32 modules support 80 MHz QIO access mode.
- You are recommended to load the second-stage bootloader in QIO mode, as QE is not set by default in some flash status registers.

5.3.28 How to configure the RMII synchronous clock for the Ethernet of ESP32?

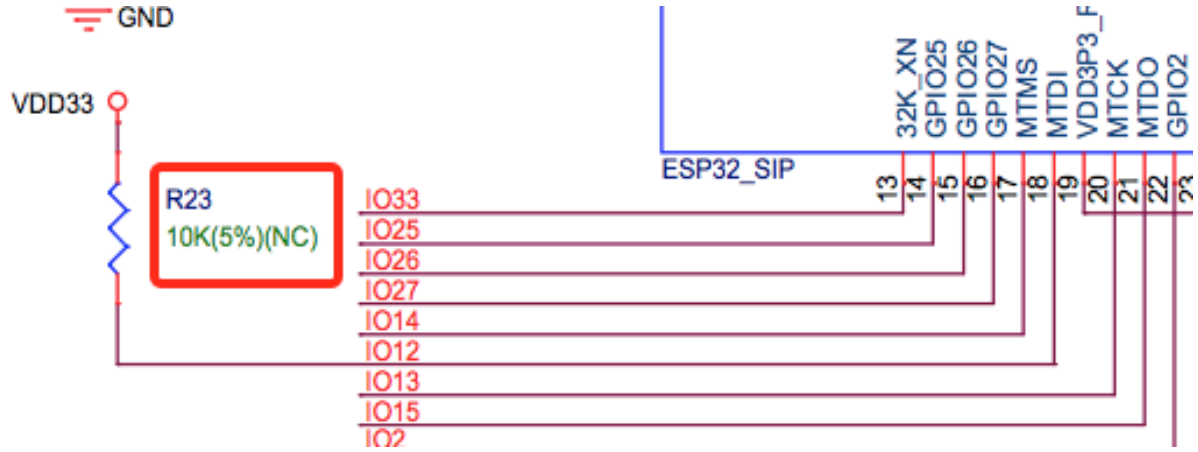
- To run a test, please refer to esp-idf/examples/ethernet/basic.
- When GPIO0 provides clock output for PHY, the Ethernet connection of the IP101 PHY chip can be unstable. Therefore, you are recommended to connect a 50 MHz crystal to PHY with GPIO0 as input.
- Because of the characteristics of GPIO0, the IO should be set to control the enable pin of PHY.
- Please read Ethernet document.
- You may also refer to 1SCH_ESP32-ETHERNET-KIT Schematics.

5.3.29 How can I hard reset ESP8266? Is hard reset active low or active high? What are the requirements for reset?

- The Pin32 EXT_RSTB of ESP8266 is the reset pin. This active low pin has an internal pull-up resistor. To prevent external factors triggering a restart, it is recommended that the EXT_RSTB cabling be as short as possible and an RC circuit be added to the EXT_RSTB pin.
- The CHIP_EN pin of ESP8266 can also be used as a hard reset pin. When you use the CHIP_EN pin as a reset pin, the reset is active low. To reset and restart ESP8266, the input level should be lower than 0.6 V and last for more than 200 µs. It is recommended to use the CHIP_EN pin for chip reset. For more information, please refer to Section *Reset* in the ESP8266 Hardware Design Guidelines.

5.3.30 What does the term NC mean in Espressif schematics?

NC is the acronym of "No Component". If you see a pull-up resistor is marked NC as shown in the figure below, it indicates that the component is not installed.



5.3.31 How can I use multiple antennas with ESP32-S2?

- Using multiple antennas with ESP32-S2 is similar to that with ESP32. You may refer to ESP32-WROOM-DA Datasheet.
- For detailed instructions, please refer to ESP-IDF Programming Guide.
- You can add an RF switch to select antennas.

5.3.32 Does ESP32-C3F SPI CS0 pin need an external 10 kΩ pull-up resistor?

- The SPI controller of ESP32-C3F supports software-programmable CS (Chip Select) pin without external 10 kΩ pull-up resistor. - In ESP32-C3F, the CS pin can be set to any GPIO pin via SPI controller configuration. The GPIO state can be set in the code to control the level of the CS pin. When the SPI bus is idle, the CS pin is automatically pulled up to the default state of the GPIO pin without an external pull-up resistor.
- Please note that when using a software-programmable CS pin, to select the target device, the pin should be manually pulled down before the SPI bus transmission. After the transmission is completed, pull the CS pin high to release the device. Additionally, the level and status of the CS pin should be adjusted according to the actual situation to ensure the stability and reliability of the SPI bus.

5.3.33 Is there any hardware design reference for ESP-Skainet Speech Recognition?

Please refer to ESP32-Korvo V1.1 User Guide.

5.3.34 Is it necessary to connect a 32 kHz RTC crystal to the ESP32 chip in hardware design?

- An external 32 kHz RTC crystal is optional, not mandatory. Its main purpose is to provide higher time accuracy and reduce power consumption. Typical application scenarios include:
  - Low Power Bluetooth (BLE) Design: An external 32 kHz crystal is mainly used for timing during BLE Light-sleep. If a 32 kHz external crystal is not added to the hardware, Light-sleep may not function correctly in some BLE applications.
  - Improved time accuracy: Compared with the internal RC oscillator, an external crystal oscillator is less affected by the environment and has higher accuracy. This is crucial for applications that require periodic wakeups to receive beacons (such as Wi-Fi/Bluetooth keep-alive), as it helps prevent RF receive windows from being extended due to clock drift, thereby significantly reducing average power consumption.
- If the application scenario does not involve low power mode or there is no need to maintain high precision timing during sleep, the external crystal can be omitted.
- The ESP32 integrates a 150 kHz RC oscillator and a clock derived from an 8.5 MHz oscillator divider, which can serve as RTC clock sources. Although their frequency stability is more sensitive to temperature variations, they can replace the external 32 kHz crystal oscillator in applications that do not have strict accuracy requirements.

5.3.35 For the ESP32-MINI-1 module, is there a component library for Altium Designer?

- Our hardware schematics are developed with PADS. To find the .asc file that can be converted and opened in Altium Designer, please go to ESP32-MINI-1 Reference Design.
- For hardware reference designs of other modules, please refer to technical documents.

5.3.36 Can I change the input voltage of UART0 of ESP8266 from 3.3 V to 1.8 V?

Yes. VDDPST is the power domain for UART0, the input voltage of which can be 1.8 V theoretically.

5.3.37 Is the level of UART0 of ESP8266 determined by VDD (VCC_WIFI) or VDDPST (VCC_CODEC_IO)?

The digital power voltage is determined by VDDPST, so the level of UART0 of ESP8266 is determined by VDDPST (hardware power domain).

5.3.38 What should be noted when I connect an external PSRAM to ESP32-D2WD?

- Please enable `CPU frequece 240 Mhz` and `RTC clock 80 Mhz` as follows:
  - `menuconfig > Serial flasher config > Flash SPI Speed (80 Mhz)`
  - `Component config > CPU frequency (240 Mhz)`
  - `Component config > ESP32 specific > [*]Support for external, SPI-connected RAM`
  - `Component config > ESP32 specific > SPI RAM config > Set RAM clock speed (80 Mhz clock speed)`

5.3.39 When the VDD power supply of ESP32 slowly rises from 0 V to 3.3 V, why does the chip not start as usual?

- This problem occurs because the power-on sequence requirements are not met. To start the chip, when VDD reaches 2.3 V, the EN voltage should not exceed 0.6 V.
- If the VDD rise time is too long, the RC circuit on the EN side of the chip will not be able to delay EN.
- You may modify the RC circuit, for example, increase the capacitance, adjust the resistance, or use the Reset chip to control EN state.
- When the voltage provided to ESP32 is detected to be less than 2.3 V, you are recommended to pull down the EN pin of ESP32.
- For ESP32 power-on sequence description, please refer to ESP32 Datasheet.

5.3.40 When using the ESP32-WROOM-32D module, can I set GPIO12 for other uses?

- GPIO12 is a strapping pin that controls the startup voltage of SPI flash. The SPI flash startup voltage of the ESP32-WROOM-32D module is 3.3 V, so GPIO12 needs to be pulled down during powering on.
- If you need to set GPIO12 for other uses, please use the command espefuse.py set_flash_voltage 3.3v in the esptool to set the voltage through VDD_SDIO as 3.3 V.
- It is possible to connect VDD_SDIO to 3.3 V in hardware directly without burning eFuse again.
- In the mass production stage, you can also download the firmware directly by modifying the default configuration of `ESP32_EFUSE_CONFIG` to `config_voltage = 3.3 V` in `config/esp32/utility.config` in the flash download tool.

5.3.41 When connecting an external flash to ESP32-WROOM-32D module, is it possible if I do not use GPIO6 ~ GPIO11 pins?

ESP32 has three sets of SPIs (SPI, HSPI, and VSPI), which can access the external flash through the SPI0/1(HSPI/VSPI) bus. The external flash connected to other pins (pins other than GPIO6 ~ GPIO11) can only receive data for storage, but not run code. If you need to run code from flash, please connect the flash to GPIO6 ~ GPIO11 pins only.

5.3.42 Do I need to add a shielding case to the PCB of ESP32 modules?

- Whether a shield needs to be added depends on the specific application scenarios and requirements.
  - In some demanding application scenarios, such as severe wireless communication interference environments, high electromagnetic compatibility (EMC) test requirements, etc., installing a shield can effectively reduce external interference and mutual interference on the PCB board, improving the system's stability and reliability. At this time, the shield should be made of conductive material and grounded to ensure its effectiveness.
  - On the other hand, if the application scenario is relatively simple, such as low wireless communication interference and low EMC requirements, the effect of adding a shield may not be very obvious and may increase system cost and complexity.
  - If the board has other signal interference, such as 2G, 3G, 4G, Wi-Fi, Bluetooth, Zigbee, etc., it is recommended to add a shielding case.

5.3.43 Do I must use GPIO0, GPIO1, or GPIO3 of ESP32 as the I2S CLK pin?

The MCLK pin must use GPIO0, GPIO1, or GPIO3. The other clock pins can use any GPIOs. Note that GPIO0 is generally not recommended for other functions because it is a strapping pin.

5.3.44 Does the ESP32-U4WDH chip support external PSRAM chips?

- The ESP32-U4WDH chip supports external PSRAM chips. However, only the ESP-PSRAMXXH chip released by Espressif is supported. Third-party PSRAM chips are not supported.
- For hardware design, all the PSRAM pins except for the CS pin can be multiplexed with Flash. For more information, please refer to the ESP32 Hardware Design Guidelines.
- Also, when designing the PCB, please make sure that the GND of the PSRAM to the GND of the ESP32-U4WDH is as short as possible; Otherwise, the signal quality may be affected.

5.3.45 Does ESP32 support connection to an external SD NAND flash chip (instead of the default NOR flash chip) via the SPI0/SPI1 interface for storing application firmware?

- The ESP32 chip does not support external SD NAND Flash chips using the SPI0/SPI1 (connect the core Flash) interface.
- If you want to store external data, it is recommended to use the SPI2, SPI3, or SDIO interface of ESP32 to connect to an external NAND SD chip.
- SPI2 and SPI3 can be used via any GPIOs, while the SDIO interface can only be used via the specified interface. For more information, please refer to Section *Peripheral Pin Configurations* in the ESP32 Datasheet.

5.3.46 Does it support to connect a second PSRAM chip externally based on the ESP32-S3R8 chip?

- No, it is not supported. The reasons are as follows:
  - The PSRAM chip is connected to the MSPI bus. There are only two CS signals from the MSPI peripheral, one is connected to the flash, another is connected to the PSRAM.
  - CPU accesses external memory via cache and MSPI. A GPSPI peripheral is not accessible cache.

5.3.47 Could you please provide the 3D model and Footprint files of the ESP32-S3-WROOM-1 module?

The 3D models and Footprint files for the modules are available under the espressif/kicad-libraries library.

5.3.48 Does ESP32/ESP32-S2/ESP32-C3/ESP32-S3 support powering the RTC power domain only to keep the chip working with low power consumption?

Not supported. For detailed explanation, refer to the hardware design instructions in the document RTC Clock Source (Optional) (ESP32-C3).

5.3.49 How can I improve the EMC performance?

- At the hardware level, the following measures can be taken to improve the EMC performance of the PCB board.
  - The EMC performance with a four-layer board design will be better than a two-layer board hardware design.
  - Add filtering circuits to the power supply circuit.
  - Add ESD or magnetic beads to the antenna circuit.
  - Add a zero-ohm series resistor to the SPI Flash communication lines to lower the driving current, reduce interference to RF, and adjust timing for better interference shielding.
  - Keep GND intact as much as possible.
  - For more hardware design suggestions, please refer to 《ESP Hardware Design Guidelines》 .

5.3.50 Why do I need to connect a 499 Ω resistor to U0TXD for ESP32-S3?

The 499 Ω resistor is reserved for the U0TXD to suppress 80 MHz harmonics. For more information, please refer to UART.

5.3.51 How to calibrate the ESP32-S3 ADC in hardware?

ESP32-S3 already has the ADC calibrated in hardware on the chip. ESP32-S3 ADCs can be sensitive to noise, resulting in large differences in ADC readings. Depending on the usage scenario, you may need to connect a bypass capacitor (e.g. 100 nF ceramic capacitor) to the ADC input pads for minimizing noise. In addition, multi-sampling can be used to further mitigate the effects of noise.

5.3.52 How to design an automatic download circuit based on the ESP32 series chip?

You can refer to the hardware design of the automatic download circuit in the ESP32-DevKitC V4 schematic.

5.3.53 Which crystal oscillator should be used on the ESP8266 chip?

The ESP8266 chip requires the 26 MHz crystal oscillators to start the chip. The crystal precision should be ±10 PPM. For details, please refer to 《ESP8266 Hardware Design Guidelines》 .

5.3.54 Do the ESP32-C2, ESP32-C3 and ESP32-C6 chips support external PSRAM chips?

ESP32-C2, ESP32-C3, and ESP32-C6 do not support external PSRAM chips. However, ESP32-C61 will support 2 MB PSRAM.

5.3.55 When the ESP32-C3 is powered by a battery, it may fail to start if the supply voltage gradually decreases, for example, when the battery is fully discharged and then recharged. In such cases, the solution could only be disconnecting the battery from the ESP32-C3 and reconnect a fully charged battery, or to connect a voltage regulator diode between the 3.3 V pin and the EN pin to ensure the chip starts properly. What is the root cause of this situation? Is there an optimal solution?

- Root cause: When powering up and resetting the ESP32-C3 chip, the CHIP_EN pin needs to meet the power-up timing specifications outlined in the ESP32-C3 datasheet or ESP32-C3 hardware design guidelines. If the battery discharge and power-up process are relatively slow, ESP32-C3 may not be fully reset, resulting in certain units of the chip being in an uncertain state.
- Solution: Currently, if using battery power or energy storage systems, this issue can be addressed by adjusting RC component values, using voltage divider circuits with two resistors, or using a reset IC, which is a more commonly used approach. For detailed information regarding RC component values and related resistors, please refer to ESP32-C3 Family Hardware Design Guidelines.

5.3.56 Does the 3x3 GND grid on the ESP32 series module need to be copper-plated?

It is recommended to plate copper on the 3x3 GND grid of the module.

5.3.57 Why does ESP32 fail to start upon first power-up, and it requires a reset to start normally?

Please check if the Boot pin is connected to a capacitor that is too large.

5.3.58 What could be the possible hardware-level reasons for the failure of Ethernet PHY initialization?

It is suggested to try to use a 0 Ω series resistor.

5.3.59 Ethernet PHY initialization is successful, but unable to obtain an IP, what could be the possible reasons at the hardware level?

Please check if there is a significant load on GPIO0, you can remove the additional circuit and test again.

5.3.60 What could be the reason that a PCB designed with the ESP32 chip can not connect to the router properly?

- Please connect a 24 nH inductor in series on XTAL_P.
- Please adjust the capacitors on both sides of the crystal oscillator according to these steps.

5.3.61 Is it possible to change the default power-up reset initial state of GPIO6 (JTAG pin MTCK) of ESP32-C6, which is input enabled and internal weak pull-up resistor enabled (IE & WPU), to input enable only (IE) by burning eFuse?

- Yes. After setting EFUSE DIS_PAD_JTAG to 1, the power-up reset initial state of GPIO6 (JTAG pin MTCK) is changed to input enabled (IE). Please refer to *Section 2.2 Pin Overview* in ESP32-C6 Datasheet.
- The EFUSE DIS_PAD_JTAG can be set to 1 using the espefuse.py burn_efuse DIS_PAD_JTAG command.
- You can also add the following code to the application code to set EFUSE DIS_PAD_JTAG to 1:

```
#include "esp_efuse.h"
#include "esp_efuse_table.h"

esp_efuse_write_field_bit(ESP_EFUSE_DIS_PAD_JTAG);
```

5.3.62 Can the VBAT pin of ESP32-H2 be powered independently?

In theory, RTC is supported for power supply, but 3.3V is already connected inside the module, so this pin is actually not available.

5.3.63 Is ESP32-P4 VBAT (Backup Battery) now fully supported?

IDF has support for VBAT, please refer to the `examples/lowpower/vbat` example. The document under solution is outdated, please refer to the latest IDF document. VBAT power supply is only switched during deep sleep, and light sleep does not switch to VBAT power supply. In this case, a 32.768 kHz crystal oscillator must be externally attached.

5.4 RF Related

5.4.1 Does the RF performance of an ESP32 module degrade if it runs with a 2.8 V supply?

Yes, its RF performance may become unstable. It is recommended to follow the suggested operating voltage range specified in the module's datasheet.

5.4.2 What are the modulation methods supported by ESP chips?

- ESP8266 supports BPSK/QPSK/16QAM/64QAM/DBPSK/DQPSK/CCK.
- ESP32 supports BPSK/QPSK/16QAM/64QAM/DBPSK/DQPSK/CCK/GFSK Π/4-DQPSK 8-DPSK.
- ESP32-S2 supports BPSK/QPSK/16QAM/64QAM/DBPSK/DQPSK/CCK.
- ESP32-C3 supports BPSK/QPSK/16QAM/64QAM/DBPSK/DQPSK/CCK/GFSK.
- ESP32-S3 supports BPSK/QPSK/16QAM/64QAM/DBPSK/DQPSK/CCK/GFSK.

5.4.3 How can I get the RF related information (e.g., antenna specification, antenna pattern, etc.) for certification?

For such information, please contact Sales for help.

5.4.4 Why does ESP32 automatically reduce its transmit power when it uses the RF Test Tool at 80 °C?

Temperature compensation is disabled by default when ESP32 runs the fixed frequency firmware. Therefore, the power reduces at a high temperature. To enable temperature compensation, please send `tx_pwr_track_en 1 1 0` to ESP32 through the default log serial port.

5.4.5 How to improve the receiving distance and strength of Wi-Fi signals for ESP32-WROVER-E? (Application scenario: Wi-Fi probe)

- In terms of software, you can either set the maximum transmit power by API esp_wifi_set_max_tx_power(), or set that via menuconfig: `Component config` > `PHY` > `Max Wi-Fi TX power(dBm)`. The default maximum transmit power is 20 dBm.
- If the transmit power has been set to the maximum value, you may improve the efficiency of the antenna and receiving devices as follows:
  - Adjust the module direction so that the stronger radiation direction of the antenna points to the receiving device, realizing the farthest radiation distance.
  - Make sure there is no metal or blocking object near the antenna of the module, no PCB on the back of the antenna, and the Wi-Fi signal is not interfered by other signals of the end device.
  - Use the IE series module with an antenna connector if the performance of the PCB antenna cannot meet requirements, so that an external antenna with higher directional gain can be connected.
  - Increase the radiation efficiency of the antenna in the receiving device.

5.4.6 How to write phy_init data to flash ?

:CHIP: ESP32 :

To write the phy_init data into flash, you need to first check the `menuconfig` > `Component config` > `PHY settings`. Depending on the configuration, the flashing method differs.

- If `CONFIG_ESP_PHY_INIT_DATA_IN_PARTITION` is not enabled, the phy_init data will be directly embedded into the application binary file, and does not need to be flashed separately.
- If `CONFIG_ESP_PHY_INIT_DATA_IN_PARTITION` is enabled, the phy_init data must be flashed separately to a phy data partition. The steps are as follows:
  - The partition table of the project must include a phy data partition, for example:

```
# Name,   Type, SubType, Offset,   Size, Flags
phy_init, data, phy,     ,        0x1000,
```

  - After building the project, a separate `phy_init_data.bin` file will be generated.

    Note: You can also use the PowerLimitTool tool to generate the `phy_init_data.bin` file.

  - Use the flash download tool to flash the generated `phy_init_data.bin` into the flash at the address specified in the partition table.

Note: If `CONFIG_ESP_PHY_INIT_DATA_IN_PARTITION` is enabled, but the `phy_init_data.bin` file is not actually flashed into the phy partition, an error will occur.

5.4.7 How can I optimize the second harmonic and other spurious signals created by my own products?

The second harmonic mainly comes from the radiation generated by the RF link and PA (power amplifier) power supply. The backplane (board size) and the product also make impact on the second harmonic. Therefore, it is recommended to:

- Add a ground capacitor of approximately 2.4 pF to the RF matching circuit to reduce the spurious radiation coming from the RF link.
- Add a series inductor to the PA power supply (Pins 3 and 4 of ESP32) to reduce the spurious radiation coming from it.

5.4.8 How can I suppress the harmonics of 80 MHz?

If the harmonics of 80 MHz (160 MHz, 240 MHz, 320 MHz, etc.) exceed the allowable levels of spurious emissions, you can add a resistor of approximately 470 Ω to the data transmission circuit (TXD) to suppress those harmonics.

5.4.9 Is manual power calibration required for the Espressif modules that connect to external antennas?

No, it is not required. When an external antenna is used, please make sure it is connected properly before the module is powered up. Then, the module performs a self-calibration, including power calibration.

5.4.10 What is the default duty cycle for the "default" level when using the "RF Test Tool" to set the Wi-Fi "TX continues" mode?

- The default duty cycle for the "default" level is 98% or higher, and cannot be modified.

5.4.11 Does the ESP RF Test Tool support single carrier testing?

- Yes. To do the single carrier testing, you can set the `Test mode` to `TX tone` in the test interface.
- Before using ESP RF test tools, it is recommended to read the guide RF Test Items.

5.4.12 Is there a corresponding software API for the ESP RF Test Tool to implement the relevant RF test functions in the application firmware?

Please refer to the esp-idf/examples/phy/cert_test example, which demonstrates how to use the API for certification testing.

5.4.13 Is it feasible to design a 50-ohm impedance on an FPC?

Although theoretically feasible, it places high demands on the FPC, and there could be risks in EMC handling. Therefore, it is not recommended to place a chip directly on an FPC.

5.4.14 Can the Espressif U series antenna modules with IPEX antenna connectors (such as ESP32-C3-WROOM-02U) be directly connected to SMA antennas?

No, this is not supported. It is suggested to use an IPEX antenna connector directly. If you need to connect an SMA antenna, please use an IPEX to SMA adapter cable.

5.4.15 Can I reuse Espressif's FCC/CE certification reports for the ESP32-S3-WROOM-1 module without additional fixed-frequency testing?

Some data can be reused, but EMC testing is still required for the final product.

5.4.16 What could be the reason for the signaling test results being worse when provisioning with Bluetooth first, then testing Wi-Fi after turning off Bluetooth, compared to testing Wi-Fi alone?

Bluetooth does not affect the actual performance of Wi-Fi, but it does affect the signaling test process. If Bluetooth still occupies radio frequency front-end resources during Wi-Fi testing, Wi-Fi will drop packets, which in turn affects the interaction between the instrument and the chip during the signaling test process, causing the measured data to deviate, rather than the actual RF performance decreasing. It is necessary to ensure that Bluetooth is completely turned off (thoroughly deinitialized) during testing.

5.5 Process and ESD Protection

5.5.1 What should be paid attention to during the ESP32 ESD test?

- The Electrostatic Discharge (ESD) test for ESP32 is conducted to ensure that the device has sufficient tolerance to withstand electrostatic discharge. The precautions are as follows:
  - ESD testing should be conducted in an ESD laboratory or ESD protection area, which should have good grounding protection and electrostatic discharge protection facilities.
  - When conducting ESD testing, please use ESD testing equipment that meets international standards, including ESD generators and ESD grounding mats, to ensure the accuracy of the test results.
  - When conducting ESD testing, please make sure you are using a stable 3.3 V voltage. If the EN trace is too long, it may cause a reboot.
  - ESP32 devices should be tested multiple times to verify the reliability of their tolerance, and the test results should be recorded and analyzed.
  - If the module does not respond, please check the voltage of the air discharge or contact discharge used in the test.

5.6 Production Test

5.6.1 Why can some modules download firmware normally when using DIO/DOUT, but encounter program abnormality when using QOUT/QIO?

- Firstly, please check the modes supported by flash in the module and whether the module routing meets the requirements of modes.
- Secondly, please check the QE bit of the status register of flash, which determines whether the flash supports the QUAD mode or not.
- Different ESP chips/modules use flashes from different manufacturers. Some flashes have QE disabled by default. Thus, it is necessary to check whether the flash supports Quad mode through actual testing.
- When ROM boots a second stage bootloader, the secondary read will fail if the configuration parameters are read in the QIO mode because the QE bit is disabled.
- It is recommended to program firmware in the DIO mode and to configure the QIO mode in `menuconfig`. The configuration enables the QE bit in the second stage bootloader and then boots the app bin to use the QUAD mode.

5.6.2 How to get the production test tool?

CHIP: ESP32 | ESP8266

- Please click production test tool to download.

5.6.3 When I use the `esptool.py burn_custom_mac` command to write the user-defined MAC address, why is the MAC address read by the `esptool.py read_mac` command still factory default?

- The `esptool.py read_mac` command can only read the MAC address written in eFuse BLOCK0 by default, but the user-defined MAC address written with the `esptool.py burn_custom_mac` command is in eFuse BLOCK3. You may use the `espefuse.py get_custom_mac` command to check the MAC address written to eFuse BLOCK3.
- For more information, please refer to esptool documentation.

5.6.4 When downloading bin files to ESP32-WROVER-E (16 MB flash) using Flash Download Tool, multiple separate bin files can be downloaded successfully, but downloading the combined firmware (12 MB) failed. Why?

Since the combined firmware is mostly "0xFF" with relatively high compression rate, the amount of data after decompression would be relatively large for the same length of compressed data, resulting in a timeout error (default 7 seconds) after a long download time. To solve this issue, in Flash Download Tool, go to `configure` > `esp32` > `spi_download`, and disable the compression configuration option as follows:

```
compress = False
no_compress = True
```

5.6.5 Where can I find the MTBF report for a chip or module?

MTBF (Mean Time Between Failures) values are not publicly provided. MTBF can vary widely depending on test scope, operating conditions, and actual application scenarios. Publishing a single number may cause misunderstanding or lead to incorrect conclusions for real-world use.

Chapter 6

Test verification

6.1 Power Consumption Verification

6.1.1 Why does ESP32 reboot when it is woken up from Deep-sleep mode?

When ESP32 is in Deep-sleep mode, the digital core is powered off and the information stored in CPU will be lost. After ESP32 is woken up from Deep-sleep mode, it re-boots firmwares and re-loads them to the internal memory. The application information that requires to be reserved can be saved in RTC, as RTC is still powered on in Deep-sleep mode. The reserved information can be loaded after wake-up.

6.1.2 What sleep modes does ESP32 support? What is the difference between them?

ESP32 supports three sleep modes: Modem-sleep, Light-sleep, and Deep-sleep.

- Modem-sleep: CPU works normally and the clock is configurable. The station (ESP32) automatically turns on after it is connected to the AP. After ESP32 enters Modem-sleep mode, the RF module is shut down, and the station remains connected to the AP. If ESP32 disconnects to the AP, it will not work in Wi-Fi Modem-sleep mode. In Modem-sleep mode, the CPU clock frequency can be lowered to further reduce the current consumption.
- Light-sleep: CPU is suspended and the digital core clock is limited. When ESP32 is in Light-sleep mode, not only the RF module is closed, CPU and partial system clocks are also suspended. After ESP32 exits Light-sleep mode, the CPU resumes working.
- Deep-sleep: The digital core is powered off and the information stored in CPU is lost. After ESP32 enters Deep-sleep mode, all modules are closed except for RTC. After it exits Deep-sleep mode, the entire system restarts, which is similar to the system reboot. ESP32 does not remain connected to the AP in Deep-sleep mode.

Please refer to *Table 8: Power Consumption by Power Modes* in ESP32 datasheet for the corresponding sleep power consumption.

6.1.3 Can ESP32 in Deep-sleep mode be woken up by any RTC_GPIO?

Yes. For the configuration of RTC_GPIO, please refer to ESP32 datasheet > Chapter *Pin Definitions* > Section *Pin Description*.

6.1.4 What is the power consumption of ESP8266 when the CHIP_PU pin is at the low level?

- CHIP_PU pin is the module EN pin. When the pin is set to the low level, the power consumption of the chip is about 0.5 μA.
- In Table Power Consumption by Power Modes of ESP8266 Datasheet > *Functional Description* > *Power Management*> *Table 3-4. Power Consumption by Power Modes*, shut down power mode means CHIP_PU is pulled down and the chip is disabled.

6.1.5 Why does the minimum current of ESP32 in Light-sleep increase when the timer is not used as a wakeup source?

- By default, to avoid potential issues, esp_light_sleep_start functions will not power down flash. This is to prevent errors that may be caused if the flash is not fully powered off and back on when the device has just gone to sleep and is immediately woken up.
- For the issue details and on how to optimize power consumption in this scenario please refer to Power-down of Flash in the ESP-IDF Programming Guide.

6.1.6 In ESP32' s Deep-sleep mode, using an internal 150 KHz RTC clock or using an external 32 KHz, which consumes more power?

- If the RTC clock source is external 32 kHz crystal, there is no difference in power consumption.
- If an external 32 kHz oscillator is connected to the hardware, the power consumption will increase by 50 to 100 μA regardless of which RTC clock source is selected.

6.1.7 What are the requirements for CPU frequency to ensure normal operation of the RF module when reducing power consumption by reducing the CPU frequency?

CPU frequency should be 80 Mhz at least.

6.1.8 When I run the light-sleep example on ESP32-S3 modules, if I only use the GPIO wake-up source and do not enable the timer to wake up, the power consumption is 3 mA, which is significantly different from that on the datasheet. Why?

- When using the RTC GPIO wake-up source, please add the following code for testing before the module enters light-sleep mode. **However, please do not wake up the module immediately after the it falls asleep when only using the GPIO as the wake-up source. This is because flash may cause a failure when the duration between power-up and power-down is too short under the situation without configuring the timer.**

```
esp_sleep_pd_config(ESP_PD_DOMAIN_VDDSDIO,ESP_PD_OPTION_OFF);
```

6.1.9 When using Timer for wake-up based on the esp-idf/examples/system/deep_sleep example, although the wake-up time is set to 2.5 hours, it wakes up around 1 hour. What could be the reason?

The code block is as follows:

```
const int wakeup_time_sec = 9000;
printf("Enabling timer wakeup, %ds\n", wakeup_time_sec);
ESP_ERROR_CHECK(esp_sleep_enable_timer_wakeup(wakeup_time_
↪sec * 1000000));
```

- An overflow may occur during the `ESP_ERROR_CHECK(esp_sleep_enable_timer_wakeup(wakeup_ti * 1000000));` calculation. You may modify the code as follows:

```
const uint64_t wakeup_time_sec = 9000;
printf("Enabling timer wakeup, %lld\n", wakeup_time_sec);
ESP_ERROR_CHECK(esp_sleep_enable_timer_wakeup(wakeup_time_sec␣
↪* 1000000));
```

- Or directly write it as `esp_sleep_enable_timer_wakeup(9000 * 1000000ULL);`.

6.1.10 After enabling the Ethernet function based on the ESP32-WROVER-E module, the chip cannot enter the Power Save (Auto Light-sleep) mode. What is the reason?

Currently, ESP32 does not support using Ethernet as a wake-up source. Therefore, when Ethernet is enabled, the chip must remain in Active mode to receive data.

Chapter 7

Commercial FAQ

7.1 Which certificates have your products passed?

Please check our Certificates , where you can get all the relevant information about our products.

7.2 Does your company have the ISO Quality Management System Certification?

Yes, our company has passed the ISO9001:2015 Quality Management System Certification.

7.3 Do your chips and modules have environmental certificates such as REACH, ROHS, etc?

Our chips and modules comply with REACH, ROHS, Prop65 and many other environmental certification standards. To find out more about them, please contact our business support team by submitting a Sales Questions electronic form, where you should mention the specific environmental certificate you need to check.

7.4 Do you have distributors in China, Europe, the United States and Canada?

To get specific information about our worldwide distributors, you should contact us by filling in the required information on our Sales Questions webpage. Then, our business support team will contact you as soon as possible, giving you all the information you need.

7.5 How can I start a distribution business with Espressif?

If you are interested in becoming one of our distributors, please send your company information to: sales@espressif.com.

7.6 Where can I find your product information? Which of your products are in mass production?

You can get the basic information on our products by clicking here . If you are looking for the technical documents of our products, please click here .

7.7 Do your products have a longevity commitment?

Yes, Espressif provides a minimum longevity commitment of at least 12 years for all the products listed here .

7.8 Where can I find the SPQ (Standard Pack Quantity) and MOQ (Minimum Order Quantity) for your products?

For the standard packaging quantity (SPQ) and minimum order quantity (MOQ) of each product, please refer to the ESP Product Selector Tool.

7.9 What is your recommended purchasing method?

If you need to make a bulk purchase, please go to our Sales Questions and fill in the required information. Then, our business support team will contact you as soon as possible. If you just want to buy samples, please click here to check the corresponding purchasing method.

7.10 What's the price for bulk purchasing? How can I purchase in bulk?

Please go to our Sales Questions and fill in the electronic form you will find there. Then, our business support team will contact you as soon as possible.

7.11 Where can I find all the differences between your products (e.g. in terms of series and types)?

Please click here to find some introductory information on our products. For detailed information, please contact our sales team by clicking here .

7.12 Do your products have firmware? Can I customize my module/chip flash before the product leaves the factory? How much does this process cost? How long does it take? How can you help me do this?

Espressif Systems has developed a set of AT commands that can be used for Espressif products to easily interface with other products. All Espressif modules come with empty firmware by default from the factory. For more information, please go to our Sales Questions and fill in the required details. Our business support team will contact you as soon as possible. Additionally, in order to simplify and shorten our customers' manufacturing process, we also provide customized manufacturing services. You can go to our Manufacturing Services and check all the available flash projects. For more information, please go to our Sales Questions and fill in the required details. Then, our business support team will get in touch with you as soon as possible.

7.13 Which of your products support HomeKit? Where can I get the Espressif HomeKit SDK?

You can refer to the Espressif HomeKit SDK . Please note that the Espressif HomeKit SDK is available to MFi licensees only, and you need to provide your Account Number (6 digits) for verification purposes, when requesting the SDK .

7.14 What is your company's address?

Espressif Systems (688018.SH) is a public multinational, fabless semiconductor company established in 2008, with offices in China, Singapore, the Czech Republic, India and Brazil. Please click here to check the details of Espressif's global offices.

7.15 How can I contact your technical team?

Please visit Technical Inquiries, and let us know about any issues or confusion you're experiencing. We will assist you in resolving them.

7.16 How can I get in touch with your company?

In order to better understand your questions and needs, please click here and fill in the required information. Then, we will get in touch with you as soon as possible.

7.17 How can I tell if an Espressif module is in mass production or an NPI product?

On each Espressif module, you can find a specification identifier (4 digits to 9 digits) in the lower left corner of the module marking. All Espressif mass production modules have a specification identifier starting with XX or Mn (n can be 0, 1, 2, ⋯. for example, M0). The rest of them are NPI products. Find more information about Espressif Module Packing Information here.

Also please note that NPI product can be different from the final mass production product and may experience hardware or software issues. Thank you for your interests in our new products and helping us providing better products. Please contact our technical support team if you have any questions using our NPI products.